

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1473
1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. (Continued)

adaptation algorithm with associated maneuver detection logic was developed and favorably compared with more conventional adaptation methods. A Kalman filter to handle serially correlated observation error (without state vector augmentation) was found, restructured to improve the computational efficiency and exercised to determine parametric sensitivity to correlation effects. Prefiltering, or data compression techniques, were studied and found to significantly reduce required computation with negligible degradation in performance. Square root covariance propagation (in single precision) was found to be considerably more efficient (by a factor of 4.5) than double precision covariance for the particular filter model and computer for this application. The three-dimensional filtering problem was approached by first developing the optimal nonlinear filter as a standard and then evaluating on a relative basis several suboptimal linearized versions. It was determined that the uncoupled linearized Cartesian filter could be implemented efficiently with minimal degradation of performance from the optimal. Suggestions for future work in each of the above areas are considered and summarized.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACKNOWLEDGMENT


The author wishes to acknowledge the support, both technical and otherwise, received from individuals too numerous to list entirely. Several young professionals in the Junior Professional Development Program and regular programs contributed by performing special studies designed to answer specific questions. The more prominent of these include: Mark Hall--the three-dimensional sensitivity study; Nancy Lindsay--sensitivity studies, target modeling, and square root covariance; Ray Pajer--early adaptation work; Bill Dineen--observation error effects and prediction; and Gary Welch--three-dimensional simulation. Kee Soon Chun initiated some of the early Kalman filter and polynomial target work. Particular appreciation goes to Tom Alexander, Dr. Charles Cohen, and Phil Young for taking the interest in these problems and offering valuable suggestions and advice. David Brown, my supervisor during the period this work was performed, was always very generous with his encouragement and support. Don Robinson always managed to provide needed financial support.

FOREWORD

This report presents the results of investigations conducted during the development of the filter/predictor software for the Operational Digital Computer Program, Improved Gunfire Control System MK 68.

The report was reviewed by T. M. Alexander, Jr. of the Warfare Analysis Department.

Released by:



JAMES H. MILLS, JR., Head
Combat Systems Integration Department

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENT	i
FOREWORD	ii
NOMENCLATURE	v
I. INTRODUCTION	1
II. GENERAL DESCRIPTION OF THE DISCRETE KALMAN FILTER	5
III. THE DERIVATIVE POLYNOMIAL TARGET MODEL	11
A. EQUATIONS	12
B. INITIALIZATION	14
C. CONVERGENCE PROPERTIES	17
IV. THE RANDOM ACCELERATION TARGET MODEL	24
V. ADAPTATION	36
A. DIVERGENCE AND THE BANDWIDTH TRADEOFF	37
B. RESIDUAL STATISTICS AND MANEUVER DETECTION	39
C. VARIABLE BANDWIDTH ADAPTATION	44
D. DUAL BANDWIDTH ADAPTATION	47
VI. SERIALY CORRELATED MEASUREMENT ERROR	53
A. NOISE MODEL	53
B. RESTRUCTURED KALMAN FILTER	55
C. RESULTS AND SENSITIVITY	66
VII. TECHNIQUES FOR REDUCTION OF COMPUTATIONAL BURDEN	72
A. PREFILTERS	73
B. SQUARE ROOT COVARIANCE	81
VIII. PREDICTION	86

TABLE OF CONTENTS --(Continued)

	Page
IX. THREE DIMENSIONAL CARTESIAN FILTERING	92
A. OPTIMAL NONLINEAR FILTER	95
B. COUPLED LINEARIZED FILTER	96
C. UNCOUPLED LINEARIZED FILTER	99
D. SENSITIVITY: TEST CONDITIONS AND RESULTS	100
E. POLAR AND HYBRID FORMS	103
X. SUMMARY AND RECOMMENDATIONS	104
REFERENCES	113
APPENDIX A LISTING OF COMPUTER PROGRAMS	A-1
DISTRIBUTION	

NOMENCLATURE

B	Target bearing angle (from north)
C	Test level (nondimensional)
E	Target elevation angle (from horizontal)
H	Observation matrix
K	Gain matrix
P	State error covariance matrix
Q	Process noise covariance matrix
R	Observation error covariance matrix
r	Target range
S	Square root covariance matrix
T	Jacobian of transformation
T_M	Memory length time
t	Time
\underline{v}	Measurement error vector
W	Extrapolated value of S
\underline{w}	Process noise vector
\underline{x}	State vector
x, y, z	East, north, vertical target position
\underline{z}	Measurement vector
α, β, γ	Defined Equations (4.9)
Δt	Time interval between filter cycles
δ_{jk}	Kronecker delta
$\underline{\epsilon}$	State error vector
$\underline{\xi}$	White driving noise vector
$\underline{\eta}$	Defined Equation (6.4)
θ	Sensitivity criterion
μ	Data compression factor
ν	Residual vector

ρ	Correlation coefficient
σ	Standard deviation
τ	Correlation time constant
ϕ	Transition matrix
Ψ	Measurement noise shaping matrix
Ω	Autocovariance matrix
ω	Frequency (or angular rate)

SUBSCRIPTS

ACT	Actual
C	Cartesian
f	Filter value
L	Linearized
M	Maneuver
N	Normalized
P	Polar (spherical)
p	Predicted
pf	Prefilter
1, 2, 3	Position, velocity, acceleration
3D	Three-dimensional

OPERATORS & MISCELLANEOUS

* (Superscript)	Serially correlated error case
(Hat)	Estimated
(k/j)	Conditional at time k given data through time j
(k)	At time k
E[]	Expected value

Note: A few other letters used are defined locally.

I. INTRODUCTION

This report is designed to give the reader an in-depth view of the development of an adaptive, Kalman, target tracking filter. Emphasis is placed on the synthesis and analysis so as to convey the rationale for various design decisions. Since this filter was specifically developed for the improved (digital) MARK 68 Gunfire Control System (GFCS), the parametric studies necessarily reflect values characteristic to such systems and their particular scenarios.* To the extent possible, however, the analysis has been generalized and deals in a broad manner with the problem of tracking and predicting the state of a general target for fire control or other purposes. The MARK 68 GFCS is in fact multifunctional as it must be able to track and engage targets of all types. For the most part, the various techniques used in this filter/predictor are not new and can be found scattered throughout the literature, as can be seen from the rather lengthy list of references. It is hoped, however, that this report will be a useful reference for others who will work in this area in that it combines these various state-of-the-art concepts and brings them to bear on this particular application. This document summarizes work performed to date and also indicates the directions that future investigations might take.

Let us first consider the basic functions performed by a gunfire control system. As shown in Figure 1.1, the heart of the system is the so-called predictor-ballistics loop. Essentially, the ballistics section generates the possible four-dimensional trajectories (or

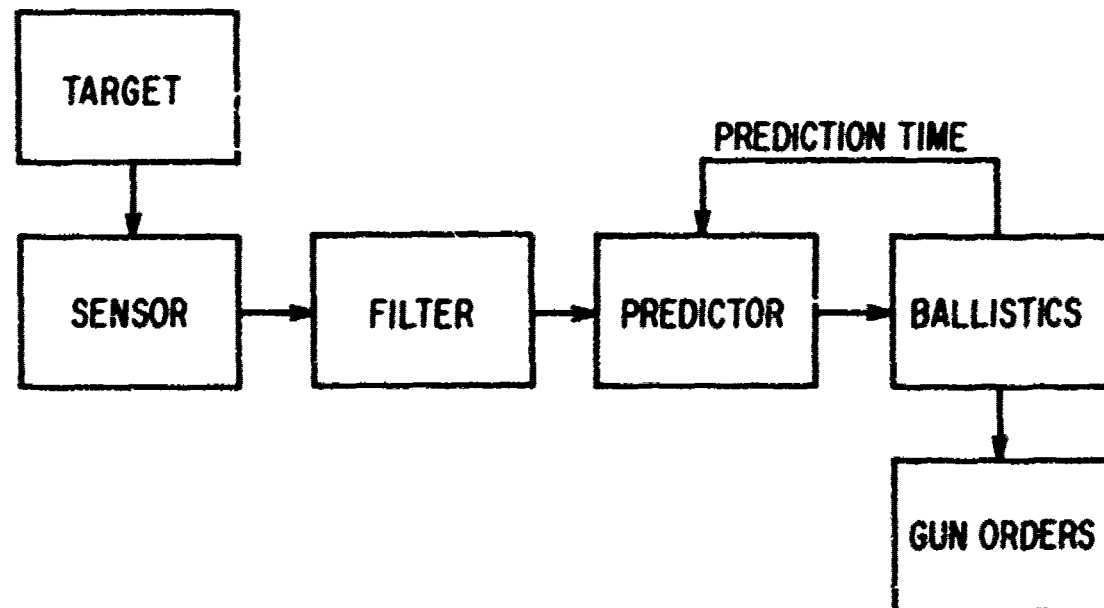


Figure 1.1. Simple Functional Diagram of a Gunfire Control System

*A classified report dealing solely with the MARK 68 GFCS implementation and performance will be published at a later date.

perhaps only terminal coordinates) of ownship's projectiles. The prediction section must supply similar information about the target trajectory. The loop is then iteratively closed by matching the spatial and temporal coordinates of the projectile to the target to effect an intercept. In this report, we will be dealing only with the "front end" of the system, i.e., from the target to the predictor. Quite obviously the predictor is the ultimate product required for the front end of the loop, and this factor will be repeatedly stressed in this report. If we had some other means of specifying the target trajectory, there would be no need for the other elements of the front end. In fact, of course the target trajectory is not directly available to the fire control system. Instead, the sensor tracks the target's current position, superimposing measurement error in so doing. The purpose of the filter then is to process the noisy position measurements in such a manner as to estimate the parameters required for the prediction model. Such parameters might include smooth current target position, velocity, and acceleration.

It is assumed for this report that measurements are made of target position only since rates (e.g. Doppler, gyros, etc.) are not currently available from GFCS sensors. It will be assumed that these measurements have been stabilized (i.e., ownship angular motion removed) and transformed to Cartesian coordinates. The implications of this transformation will be considered at length later in the report. The use of stabilized coordinates, however, is probably universal since one does not want the target motion filter to have to deal with ownship angular motion. The origin of the coordinate system shall be a reference point on ownship and common to all sensors. Tracking, filtering, and prediction will therefore be performed in ownship coordinates. This "ownship coordinate system" is defined simply as

$$\underline{X}_{T-O}(t) = \underline{X}_T(t) - \underline{X}_O(t) \quad (1.1)$$

where $\underline{X}_T(t)$ and $\underline{X}_O(t)$ are the motion of the target and of ownship relative to some earth-fixed or inertial reference frame.* Whenever it is necessary to transform to an inertial frame, say for purposes of ballistic computations, or to account (if necessary) for ownship maneuvers, Equation (1.1) and its derivatives would be used. The reason for this choice of Cartesian ownship coordinates for target filtering and prediction is simply that the motion of most targets in a GFCS scenario is better behaved--or more closely linear--in this system than in others. One must consider the fact that, if a hostile target is so close as to be in the range of fire of a gun system, most likely the engagement is mutual in that both ownship and target are maneuvering *relative to each other* in order to effect their own fire control strategy.

Let us now consider the nature of this problem from the point of view of error sources and constraints on the solution method. The prediction error, that is the error in determining the projectile-target intercept point, basically results from two (not necessarily independent)

*See the section on Nomenclature at the beginning of this report

sources: predictor modeling error and filtering or estimation error. Prediction modeling error occurs because, not knowing the target strategy, an incorrect functional form of the predictor is utilized. While we strive to construct a reasonable approximation for the actual prediction model of the general unknown target, we can never do this exactly. Therefore, even if we effected perfect filtering (i.e., perfect estimation of target parameters such as current position and rates), we would nevertheless suffer prediction errors due to unaccountable target maneuvers during the prediction time. Of course, even if target modeling were perfect, there would remain estimation errors due to sensor measurement noise which would be extrapolated from current time by the predictor. The dominant variable, an effect which significantly determines the relative and absolute magnitude of these prediction errors, is the prediction time. We shall see that prediction errors are magnified exponentially with increased prediction time.

Target modeling error contributes to the filter error as well as the prediction error, particularly for maneuvering targets. In general, a target will probably be maneuvering in some unknown fashion—at least as far as the GFCS is concerned—when the target is in range of ownship guns. There are several reasons for this. First, the target probably has a mission to accomplish or it would not be there. The target has a particular strategy to achieve that goal; and very often that goal and strategy are known to members of ownship crew. Due to the complexity of this information, however, such knowledge is not used (except very indirectly) by current gun-fire control systems. Examples of such targets are anti-ship missiles that terminally home on their targets or other weapons platforms (such as fighter bombers), which must maneuver in specific ways in order to solve their fire control problem. (The goal-oriented target and its possible prediction by a fire control system are the subject of another report to be published.) Another important reason for target maneuvers is to evade ownship weapons. Certainly, manned targets in range of our guns would be well aware of the principles of GFCS operation and behave accordingly. It is also probable that missiles will be made capable of evasive action either by preprogramming or by being under remote control. There are other reasons for target maneuver such as atmospheric turbulence, target sensor-induced errors, target control system behavior etc. In any case, it is not possible to fully and deterministically model the anticipated motion of a general target, and we must live with a certain amount of tracking and prediction error due to this limitation. We shall find that there are fundamental limitations on the ability to track maneuvering targets. A maneuver will subsequently be defined for purposes of this report as any target motion that tends to cause a certain filter implementation to diverge (i.e., estimation errors become large).

Several other factors affect the filter estimation error and these must each be considered. Tracking sensor accuracy is an important one. A very important matter is filter settling time. In a situation with more than one target to be engaged or when target acquisition and tracking commence late, it is mandatory that the filter settle quickly, within a few seconds. Computational constraints on such factors as data rate and model complexity must be

considered. Computer word length and real-time implementation constraints pose severe limitations on what one might hope to accomplish when dealing with on-line filtering and prediction. In short, the problem of target estimation for gunfire control application must be considered *in toto*, and in so doing, the filter designer finds himself facing many limitations, both fundamental and practical, that must be recognized if a reasonable design is to be realized. In the final analysis, judgments must be made that, on occasions, are difficult to justify with theoretical rigor, and filter design becomes somewhat an art as well as a science.

Before commencing the actual report, a brief discussion of the nature and structure of this study is in order. The investigation, to the extent possible, is primarily parametric in that the effects of individual parameters affecting performance were isolated, wherever possible, in order to assess the sensitivity of performance to variation of the parameters. It should be emphasized that many of the results presented in this report were based on studies designed to reveal the *relative* merits of one technique or parameter with respect to another and, as such, may not always be indicative of actual implemented performance when incorporated with other techniques and parameters in an actual tracking situation. The criterion for performance, or "norm," was chosen, due to the nature of the gun fire control problem, to be prediction performance. For a "nominal" Navy gun, a time of flight (or prediction time) of 10 seconds was chosen as typical for engagement of air targets and is used consistently throughout this report. The actual statistical form of the prediction error, used to evaluate the relative merits of parameters or filter formulations, depends upon the situation. Whenever practical, the error covariance approach was utilized since it essentially represents the result of a perfect Monte Carlo study (repeated simulation trials plus averaging). The error covariance approach can be used to evaluate optimal or suboptimal (if the "real world" or truth model is known) performance. There are many cases, however, when the truth model is not readily available, and simulation must be employed with root-mean-square prediction errors used as a norm. This is particularly true in evaluating a nonlinear adaptive filter against various nonlinear trajectories. Ultimately, of course, the filter must be evaluated in relatively uncontrolled circumstances using actual tracking data of real target trajectories.

Finally, let us consider the spatial problem. Measurements of target position are made in a spherical polar coordinate frame, but targets are generally not linearly described in such a frame. Therefore, a nonlinearity is going to appear in the full three-dimension problem. This consideration was postponed until the last chapter, and all sections until then discuss only one-dimensional estimation of target motion as a function of time. The reader may consider this single "channel" to be one direction of some orthogonal system to be defined later.

Since the author has noticed an occasional tendency among persons unfamiliar with Kalman filtering to regard the technique as exceedingly complex, mysterious and/or

omnipotent, he feels compelled to advise the reader that the antithesis is probably much closer to the truth. In the next section, the Kalman filter will be introduced and, throughout this report, the reader will note that performance remains limited, as it must be, by the quality and quantity of information available.

II. GENERAL DESCRIPTION OF THE DISCRETE KALMAN FILTER

In this section, the concepts and equations of Kalman filters will be very briefly reviewed in order to familiarize the reader with the notation and the types of information required to construct a filter of this type. The theoretical elegance and practical utility of this estimator have made it a popular subject for both mathematicians and engineers in recent years. For the reader who wishes to pursue the subject in greater depth, the author recommends such literature as Gelb (1974), Sorenson (1966) or one of many other textbooks that deal with estimation theory such as Morrison (1969) or Sage and Melsa (1971).^{*} We will deal here only with the discrete formulations since they are the more natural for digital computer implementation.

Before considering the equations, we might first qualitatively define a Kalman filter. The Kalman filter is a linear minimum-variance estimator. In fact, it is a recursive formulation of the minimum variance rules for combining certain *a priori* information with a sequence of observations that contain noise. We might also point out now that the required *a priori* information must usually be chosen with considerable care if one is to achieve the desired results. A Kalman filter is also a maximum-likelihood estimator (and in that sense an optimal linear or nonlinear filter) but only when the errors are uncorrelated and Gaussian. Of course, as mentioned in the Introduction, we really do not expect to operate our filter under optimal conditions due primarily to target modeling errors and also to correlated and perhaps slightly non-Gaussian measurement errors. Indeed, most applications of the Kalman filter are suboptimal. Kalman filters often are actually designed to be suboptimal. This frequently occurs when the dimension of the true state vector is so large as to render the computations impossible to effect on a selected digital computer. Of course, we strive to *approach* optimality within the imposed computational constraints and do just that when circumstances permit. The fact that we can often approach optimal filter performance is therefore one important reason for selecting a Kalman filter.

Another equally important reason is the inherent flexibility and control one has over the filtering process in order to maintain desired or expected filter performance. A significant computational advantage over certain other filtering techniques (such as a Bayesian estimator) is also enjoyed by Kalman filters due to a reduced matrix inversion burden. Singer and Behnke (1971), in one of the best papers on the subject, compare five types of filters

^{*}See References.

in terms of tracking accuracy and computational requirements when tracking maneuvering targets in a tactical application. They considered first and second order polynomial Kalman filters (to be discussed in the next section); an alpha-beta filter (a recursive least squares filter with a first order target model); a Wiener filter (steady-state Kalman filter); and a simple two-point extrapolator. The second-order Kalman filter was found to be the most accurate. In a later section, we shall also demonstrate that it is possible to significantly reduce by more than an order of magnitude--the computational burden of a Kalman filter. It is generally accepted that, under optimal conditions, a Kalman filter will out-perform other data-processing algorithms such as those based upon least squares. This factor, although important, is probably not in itself justification for implementing a Kalman filter since, under optimal conditions, almost any algorithm will perform acceptably. The real advantage of a Kalman filter is under suboptimal conditions. The Kalman filter "thinks" it knows how well it is performing and can be made to determine when it is not performing properly. Then, as just mentioned, the Kalman filter can utilize its flexibility to adjust the covariance and regain calculated performance. These concepts will be discussed in much greater depth in the section on adaption.

The target is assumed to be modeled by a linear, discrete, system model, as shown in Figure 2.1(a). The equation for the system is:

$$\underline{x}(k+1) = \phi(k+1, k) \underline{x}(k) + \underline{w}(k) \quad (2.1)$$

The system at time $t(k)$ is characterized by a state vector $\underline{x}(k)$ of dimension n . The state vector is assumed to propagate linearly according to an $(n \times n)$ transition matrix $\phi(k+1, k)$. For purposes of this presentation, no control input or known forcing function will be considered.

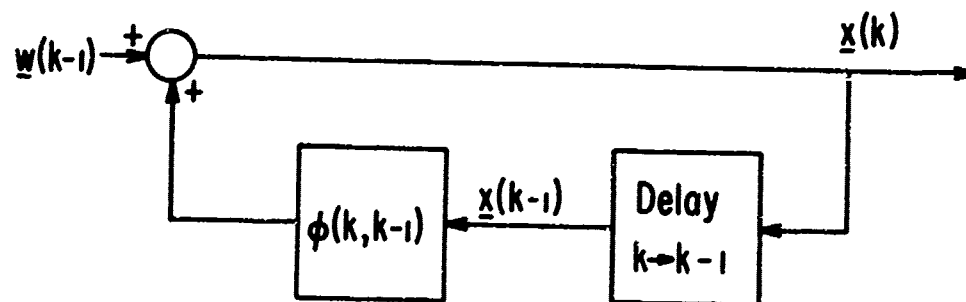
Any unmodeled effects will be assumed to be random or "nondeterministic" and are lumped into a term (the last term- \underline{w}) called "process noise." The random process noise sequence has known statistics given by

$$E \{ \underline{w}(k) \} = 0 \quad (2.2)$$

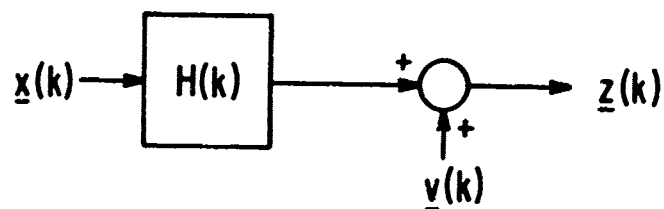
$$E \{ \underline{w}(j) \underline{w}^T(k) \} = Q(k) \delta_{jk} \quad (2.3)$$

The actual magnitude of Q is required by the Kalman filter for proper (optimal) operation. We shall see that often this information is not available *a priori* and Q must be adjusted during operation.

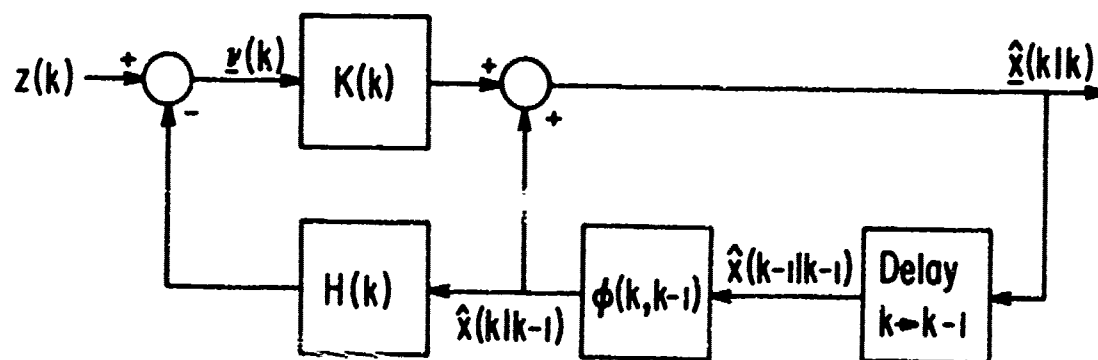
In Figure 2.1(b), we assume we have measurements $\underline{z}(k)$ of dimension m , related to the state vector by an $(m \times n)$ "observation" matrix $H(k)$ (also called the "matrix of partial



(a) LINEAR DISCRETE SYSTEM MODEL



(b) MEASUREMENT MODEL



(c) LINEAR RECURSIVE FILTER

Figure 2.1. System Flow Diagram

derivatives") corrupted with a zero-mean, Gaussian, white noise sequence $\underline{v}(k)$. That is,

$$\underline{z}(k) = H(k) \underline{x}(k) + \underline{v}(k) \quad (2.4)$$

where

$$E[\underline{v}(k)] = 0 \quad (2.5)$$

$$E[\underline{v}(j) \underline{v}^T(k)] = R(k) \delta_{jk} \quad (2.6)$$

[Notice that the use of the H matrix in Equation (2.4) readily allows the mixing of measurements of different types.]

It is further assumed that

$$E[\underline{v}(j) \underline{w}^T(k)] = 0 \quad (2.7)$$

The basic recursive, linear estimation problem is to determine an estimate $\hat{\underline{x}}(k|k)$ of $\underline{x}(k)$ that is a linear combination of the previous estimate and the current measurement as in Figure 2.1(c).

We find we can write this as

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + K(k) \underline{\nu}(k|k-1) \quad (2.8)$$

where

$$\hat{\underline{x}}(k|k-1) = \phi(k, k-1) \hat{\underline{x}}(k-1|k-1) \quad (2.9)$$

is the extrapolated estimate at $t(k)$, i.e., the estimate at $t(k)$ based on measurement data through $t(k-1)$. The *a priori* residual vector of dimension m is defined

$$\underline{\nu}(k|k-1) = \underline{z}(k) - H(k) \hat{\underline{x}}(k|k-1) \quad (2.10)$$

and $K(k)$ is the $(n \times m)$ gain matrix. The difference between the estimate and the true state is the error vector

$$\underline{\epsilon}(j|k) = \hat{\underline{x}}(j|k) - \underline{x}(j) \quad (2.11)$$

with associated $(n \times n)$ error covariance matrix defined as

$$P(j|k) = E\{\underline{\epsilon}(j|k) \underline{\epsilon}^T(j|k)\} \quad (2.12)$$

The basis of the Kalman filter is the selection of the gain matrix in Equation (2.8) that minimizes the trace of the error covariance matrix. The solution to this problem, found by

Kalman (1960, 1961), is well known and its derivation can be found in the previously mentioned references. The Kalman filter is found to be a set of recursive matrix equations that are usually written and implemented in two distinct steps. The "extrapolation" process simply predicts the state vector, using Equation (2.9), and the error covariance via

$$P(k|k-1) = \phi(k, k-1) P(k-1|k-1) \phi^T(k, k-1) + Q(k-1) \quad (2.13)$$

In this step, the error covariance usually increases since we are advancing in time without benefit of new measurements. The second step, the "update," applies to the process of reestimating the system to acknowledge that new information, $z(k)$, is available. The optimal gain is found to be

$$K(k) = P(k|k-1) H^T(k) [H(k) P(k|k-1) H^T(k) + R(k)]^{-1} \quad (2.14a)$$

$$= P(k|k) H^T(k) R^{-1}(k) \quad (2.14b)$$

Note that the bracketed term to be inverted is of dimension $(m \times m)$ which is usually smaller than the other matrices with dimension $(n \times n)$. The state can then be updated, using Equation (2.8), and the error covariance is updated as

$$P(k,k) = [I - K(k) H(k)] P(k,k-1) \quad (2.15)$$

We find that the error covariance is reduced by the update step, usually more than it was increased in the extrapolation step, causing a net reduction in the error covariance over a complete cycle of the filter (unless the filter has reached steady-state operation). If there is no process noise in the system ($Q = 0$), the error covariance terms eventually approach zero (for the models considered in this report). The error covariance significantly influences the operation of the filter through the gain matrix. In general, the larger the error covariance, the larger the gains. The gain represents the relative weighting between the old estimate and the new data. Initially the error covariance and gain are large, which represents the fact that we have not processed much data and therefore can not place much confidence in our estimate. At long times, the error covariance and gains approach their minimum values and reflect the idea that, having processed a large quantity of data, we place more confidence in our estimate. If there is no process noise, eventually the error covariance becomes small and the filter begins to virtually ignore any new data. This condition can lead to filter divergence if the dynamics are not modeled suboptimally. This problem will be discussed in a later section. A filter with small error covariance and small gains will often be referred to as having a long "memory" (or "narrow bandwidth" for those who consider filters from the frequency domain). Conversely, a filter with large error covariance and gains will be said to have a short memory or wide bandwidth.

Realization of the optimal Kalman filter, we have found, requires exact knowledge of system dynamics, measurement statistics, and process noise statistics. In general, the information is never available exactly. For example, the requirement for measurement statistics

poses some interesting problems. It is often difficult, even under controlled laboratory conditions, to obtain good error statistics for tracking instruments. It is frequently even more difficult to apply these statistics with confidence when the sensor is employed in an uncontrolled environment such as a ship at sea operating against real targets. We therefore can expect the values of the statistics assumed for the filter implementation to be slightly mismatched with the real world. Such a mismatch results in a suboptimal filter implementation. In order to assess the effect of such a mismatch, we resort to a technique known as a "sensitivity analysis," whereby we calculate the actual covariance of the suboptimal filter implementation and compare it with the optimal "real-world" value possible. If a suboptimal filter of the same structure as the optimal but with unmatched parameters (leading to suboptimal gains) is utilized, the actual error covariance P_{ACT} can be calculated with the following equations.

$$P_{ACT}(k|k-1) = \phi_{ACT}(k, k-1) P_{ACT}(k-1|k-1) \phi_{ACT}(k, k-1) + Q_{ACT}(k-1) \quad (2.16)$$

$$P_{ACT}(k|k) = [I - K(k) H_{ACT}(k)] P_{ACT}(k|k-1) [I - K(k) H_{ACT}(k)]^T + K(k) R_{ACT}(k) K^T(k) \quad (2.17)$$

The subscript ACT means that the matrix is evaluated with the true parameters. The gain $K(k)$ is assumed to be calculated with assumed design values of the parameters. An important point is that Equation (2.15) gives the actual error covariance only when the optimal gain is chosen. Strictly speaking, Equations (2.16) and (2.17) can be used only to represent the actual error covariance when a suboptimal gain is chosen but the implementation of the dynamics and measurements are assumed correct. In fact, Gelb (1974) (pages 254 and 271) argues that, under certain conditions present in this application, these actual error covariance equations are correct even with incorrect transition and observation matrices. In any case, these equations, often referred to as "simple sensitivity equations," are the equations used for all sensitivity work in this report and the more complex "model-reference" equations are not used. Presentation of sensitivity results has been found to be most recognizable if the actual parameters are held fixed and the design values allowed to vary.

Since the Kalman filter equations are recursive, it is not necessary to store a large quantity of data. This factor, along with the basic computational simplicity of the equations, allows easy implementation on a digital computer. Notice that, since the gains are not a function of the actual measurements (for a truly linear system), it is sometimes possible to precompute the gains and store them as opposed to calculating them in real time. This precomputation can be performed only when values of $\phi(k, k-1)$, $R(k)$, $Q(k)$, and $H(k)$ can be predetermined for all k . With the initial value of P specified, all future values of $P(k)$ and $K(k)$ can then be calculated. It will be shown that, for the Kalman filter application in this report, it is not possible to predetermine the above matrices since they are nonlinear functions of the actual target motion in real time. The fact that the Kalman filter must propagate the error covariance matrix as it runs is an ambiguous requirement since the filter derives both its power and its principal computational burden from these matrix equations for P .

The basic Kalman filter algorithm, for the situation described in this section (i.e., linear equations and uncorrelated noise), is summarized in Table 2.1. A FORTRAN IV

Table 2.1. Kalman Filter Algorithm

Model	$\underline{x}(k+1) = \phi(k+1, k) \underline{x}(k) + \underline{w}(k)$	(1)
Observations	$\underline{z}(k) = H(k) \underline{x}(k) + \underline{v}(k)$	(2)
Statistics	$E[\underline{w}(k)] = E[\underline{v}(k)] = E[\underline{v}(j) \underline{w}^T(k)] = 0$	(3)
	$E[\underline{w}(j) \underline{w}^T(k)] = Q(k) \delta_{jk}$	(4)
	$E[\underline{v}(j) \underline{v}^T(k)] = R(k) \delta_{jk}$	(5)
State Extrapolation	$\hat{\underline{x}}(k k-1) = \phi(k, k-1) \hat{\underline{x}}(k-1 k-1)$	(6)
Covariance Extrapolation	$P(k k-1) = \phi(k, k-1) P(k-1 k-1) \phi^T(k, k-1) + Q(k-1)$	(7)
Gain	$K(k) = P(k k-1) H^T(k) [H(k) P(k k-1) H^T(k) + R(k)]^{-1}$	(8)
State Update	$\hat{\underline{x}}(k k) = \hat{\underline{x}}(k k-1) + K(k) [\underline{z}(k) - H(k) \hat{\underline{x}}(k k-1)]$	(9)
Covariance Update	$P(k k) = [I - K(k) H(k)] P(k, k-1)$	(10a)
	$= P(k k-1) - K(k) [H(k) P(k k-1) H^T(k) + R(k)] K^T(k)$	(10b)

version of this algorithm, called KALMAN, can be found in Appendix A. This subroutine was utilized for many of the simulations required for this study.

III. THE DERIVATIVE POLYNOMIAL TARGET MODEL

In this section, the concepts and equations of a one-dimensional, derivative polynomial filter are introduced. The target model that is ultimately recommended (in Section IV) is not exactly of a polynomial form but is quite similar in implementation. As such, the parametric study of the polynomial filter in this section is applicable in many ways to the nonpolynomial target model selected in the next section and, indeed, is partly responsible for that selection. It should also be mentioned that the covariance convergence study, reported in this section, was conducted under the assumption of optimal conditions. By this, we mean the following: the actual target model is always assumed to be a polynomial matching the filter model; the measurement errors are truly uncorrelated and Gaussian; and the actual measurement error variances match the assumed (filter) measurement error variances.

Before continuing, a very brief discussion of the polynomial target model is in order. In a general tracking filter application, where the particular strategy being exercised by the target is unknown, the form of the state vector and its propagation characteristics are assumed and may or may not represent the true target motion over long periods of time. We do demand, however, the assumed target model to provide a reasonably accurate approximation of target motion for short periods of time. The choice of a polynomial, of course, is then quite logical since the ability of a polynomial to approximate a process—at least locally—is well known. Indeed, the famous Weierstrass approximation theorem tells us that polynomials can approximate any continuous* function over a finite interval to any degree of approximation desired. We are not really interested in considering very-high-order polynomials, however. It must be emphasized that the gunfire control problem is not primarily a fitting or smoothing problem but a *prediction* problem. A higher-order (and thus more accurate) polynomial fit of observed past data may not help us, and may even hinder us, to more accurately predict future target position for long extrapolation times. Also, higher order filters are computationally more burdensome and, depending on the order of the highest derivative, may not settle fast enough for the gunfire control problem (as will be shown). For these reasons, we consider polynomials with nonvanishing derivatives only through third-order (jerk). The important advantages of such low-order polynomials are computational efficiency and limited necessity for *a priori* target maneuver strategies. For a complete treatment of various polynomial filters, smoothers and predictors, the reader is referred to Morrison (1969).

A. EQUATIONS

Derivative polynomial models are described by the differential equation:

$$d^{n+1} x(t)/dt^{n+1} = 0 \quad (3.1)$$

where $d^n x/dt^n$ is the highest nonvanishing derivative and n is referred to as the "order" of the model. The state vector is defined

$$\begin{aligned} \underline{x}(t) &= [x(t) \quad dx(t)/dt \quad d^2 x(t)/dt^2 \\ &\quad \dots d^n x(t)/dt^n]^T \\ &= [x_1(t) \quad x_2(t) \quad x_3(t) \quad \dots x_{n+1}(t)]^T \end{aligned} \quad (3.2)$$

*All targets with mass must, of course, be continuous in position and velocity. Acceleration must always be finite but may be effectively discontinuous. This implies jerk may not always exist.

Notice that the state vector has dimension $(n + 1)$. In the notation of continuous linear systems,

$$d\underline{x}(t)/dt = F(t) \underline{x}(t) \quad (3.3)$$

We can write the elements of the matrix $F(t)$ as

$$f_{ij}(t) = \delta_{i,j-1} \quad 1 \leq i, j \leq (n + 1) \quad (3.4)$$

where $\delta_{i,j-1}$ is the Kronecker delta. The transition matrix is therefore

$$\phi_{ij}(k + 1, k) = (\Delta t)^{j-i}/(j-i)! \quad 1 \leq i, j \leq (n + 1) \quad (3.5)$$

where

$$\Delta t = t(k + 1) - t(k) = 1/\hat{k} \quad (3.6)$$

and

$$1/(j-i)! = 0 \quad \text{for } j < i \quad (3.7)$$

For example, for $n = 2$, we find

$$\phi(k + 1, k) = \phi = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

For purposes of this report, the data rate \hat{k} (Hertz) will always be assumed to be constant so that the transition matrix (a function of Δt only) is time-invariant. For the present, we will assume process noise is zero.

In our application, we will have available measurements of position only, so that the measurement vector and covariance reduce to scalars. That is,

$$\underline{z}(k) = [z(k)] \quad (3.9)$$

$$R(k) = [\sigma^2(k)] \quad (3.10)$$

where σ^2 is the measurement error variance. The elements of the $(m \times (n + 1))$ observation matrix are simply

$$h_{ij}(k) = \delta_{i1} \delta_{j1} \quad i = m = 1$$

$$1 = j \leq (n + 1) \quad (3.11)$$

For $n = 2$, we find the row vector

$$H(k) = H = [1 \quad 0 \quad 0] \quad (3.12)$$

The residual also reduces to a scalar quantity

$$v(k|k-1) = z(k) - \hat{x}_1(k|k-1) \quad (3.13)$$

The simplicity of the measurement matrices result in the gain calculations being effected as rather simple algebraic functions of the extrapolated error covariance. E.g., the gain is the column vector with elements

$$K_j(k) = P_{1j}(k|k-1)/[P_{11}(k|k-1) + \sigma^2(k)] \quad j = 1, \dots, n+1 \quad (3.14)$$

B. INITIALIZATION

In order to specify completely a set of equations representing a derivative polynomial Kalman filter, we have to consider a method of initialization. Since the Kalman filter equations are recursive, values of both the state and associated error covariance from the previous cycle are required. For most linear filters, the method of initialization is not particularly significant because the effects tend to vanish at large times. Indeed, that is the case for this problem. However, it is desirable to limit the excursions of the predictor while the filter is in the unconverged condition. One basically wants to form the best state estimate, based on any information that is available, that minimizes the associated initial error covariance. In the absence of any *a priori* target information, the only data available to initialize the filter will be the measurements. In order to initialize the derivative elements of the state vector, it is necessary to have available, at a minimum, enough measurements to define the required derivatives by differing techniques. This method approximates derivatives (known as "pseudo-measurements") by finite differences of current and prior position measurements. For example, let us consider the case $n = 2$ and use the well-known backward difference formulae (neglecting higher-order terms) where we approximate

$$\hat{x}_1(0|0) = z(0) \quad (3.15a)$$

$$\hat{x}_2(0|0) = [z(0) - z(-1)]/\Delta t \quad (3.15b)$$

$$\hat{x}_3(0|0) = [z(0) - 2z(-1) + z(-2)]/\Delta t^2 \quad (3.15c)$$

We can now evaluate $P(0/0)$ which we will write in the normalized form

$$P(k|k) = \sigma_e(k) \rho(k) \sigma_e(k) \quad (3.16)$$

where $\sigma_e(k|k)$ is the diagonal matrix of standard deviations of the state vector error $\sigma_i(k|k)$ and $\rho(k)$ is the matrix of normalized correlation coefficients. Assuming the measurement error variance is not rapidly changing and neglecting finite-difference approximation error, we find

$$\sigma_e(0|0) = \sigma(0) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2}/\Delta t & 0 \\ 0 & 0 & \sqrt{6}/\Delta t^2 \end{bmatrix} \quad (3.17a)$$

$$\rho(0) = \begin{bmatrix} 1 & 1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{2} & 1 & \sqrt{3}/2 \\ 1/\sqrt{6} & \sqrt{3}/2 & 1 \end{bmatrix} \quad (3.17b)$$

Unfortunately, for high-data-rate systems, the error variances of the derivatives, using the finite-difference method of state estimation, can become unreasonably large. For example, from Equation (3.17a), we find that for $\sigma(0) = 5$ yards and $\Delta t = 1/15$ second, the standard deviation of the acceleration error is approximately 292 G's! This situation is clearly undesirable since we could guess zero for acceleration and never realize an error larger than a few G's. A similar argument can be made concerning the velocity initialization but the numbers are not as dramatic. Therefore, we will choose a single-pass (using one piece of data) initialization for the second order filter by taking

$$\hat{x}_1(0|0) = z(0) \quad (3.18a)$$

$$\hat{x}_2(0|0) = 0 \quad (3.18b)$$

$$\hat{x}_3(0|0) = 0 \quad (3.18c)$$

The values for the derivative state error standard deviations would then reflect our estimates of the possible errors that Equation (3.18) would produce. This is effected as

$$P(0|0) = \begin{bmatrix} \sigma^2(0) & 0 & 0 \\ 0 & \sigma_2^2(0) & 0 \\ 0 & 0 & \sigma_3^2(0) \end{bmatrix} \quad (3.19)$$

where

$$\sigma_2^2(0) = E [dx(0)/dt]^2 \quad (3.20a)$$

$$\sigma_3^2(0) = E [d^2x(0)/dt^2]^2 \quad (3.20b)$$

are *a priori* estimates and would depend upon the type of target. For example, it has been found that $\sigma_2(0) = 300$ yards/second (approximately Mach 1) and $\sigma_3(0) = 10$ yards second² (approximately one G) quite adequately span the spectrum of possible air targets in the current target scenario. Singer (1970) has suggested a method of estimating $\sigma_3(0)$ as

$$\sigma_3^2(0) = \frac{A_{\max}^2}{3} [1 + 4 \cdot \text{Probability}(|x_3| = A_{\max}) - \text{Probability}(x_3 = 0)] \quad (3.21)$$

where A_{\max} is the maximum acceleration capability of the target which, presumably, one might estimate for a particular class of targets. The values of $\sigma_2(0)$ and $\sigma_3(0)$ for surface targets would undoubtedly be much smaller. For example, we might choose $\sigma_2(0) = 14$ yards/second (about 25 knots) and $\sigma_3(0) = 2$ yard second² (about 0.2 G). Upon comparing error-covariance results of the one-pass initialization with those of the finite difference initialization, it was found that, after approximately 10 seconds, there was no appreciable difference. On the other hand, the one-pass initialization yields considerable lower error during this initial period. For the case of $n = 3$, when a value for $\sigma_4(0)$ is required, we must estimate the maximum rate of change of acceleration. For a high-performance air target, such as a modern fighter aircraft, the author learned (from discussions with pilots) that the maximum *steady-state* turning rate over the entire speed/maneuverability spectrum is approximately 20 degrees/second ($\omega_m = 0.349$ radians/second). A value of

$$\sigma_4(0) = \omega_m \cdot \sigma_3(0) \quad (3.22)$$

was therefore chosen for initialization of the third order filters.

In conclusion, the one-pass initialization is very simple to implement as it requires no additional measurements or logic. It was found, both from error, covariance results and Monte Carlo simulation, to yield improved estimates during the initial transient phase. It is

used in all remaining work in this report and was subsequently implemented in the digital MARK 68 GFCS.

C. CONVERGENCE PROPERTIES

Early in the course of this investigation, it was discovered that fundamental information on factors that directly affect filter convergence and settling time--and therefore filter performance--was not available. Qualitative relationships were usually known, or at least suspected, between convergence rates and such variables as polynomial order, data rate, initialization methods, etc. It was felt, however, that quantitative measures of such effects were necessary if intelligent decisions and tradeoffs were to be made concerning the basic operating conditions of the filter. As mentioned previously, these studies were conducted under optimum operating conditions and, as such, are intended to reflect relative performance of one filter/parameter set with respect to another.

The two significant parameters that describe a polynomial filter are the polynomial order n and the data (or cycle) rate

$$dk/dt = \dot{k} = 1/\Delta t$$

In order to establish the relationships between filter performance and these parameters, the filter matrices of the preceding paragraphs were inserted into the general Kalman filter subprogram in Appendix A which was exercised for various values of these parameters. Specifically, the values $n = 1, 2$, and 3 , corresponding to constant-velocity (first-order), constant-acceleration (second-order) and constant-jerk (third-order), were selected. Data rates of $\dot{k} = 2, 4, 8, 16$, and 32 Hertz were considered to span the range of reasonable possibilities. Assuming a typical constant measurement error standard deviation and zero-process noise Q , the error covariance and gains can be calculated independently and no target measurements need be simulated. All conditions were assumed to be optimal so that filter error covariance would truly represent filter convergence. In keeping with the idea, previously expressed, that predicted position is the important quantity, the quantity actually used by a fire control system, the standard deviation of predicted position error $\sigma_1(t + tp|t)$ was chosen as the ultimate measure of relative convergence. Predicted position, of course, can not settle until all elements of the state vector have settled so that $\sigma_1(t + tp|t)$ is an effective "norm" of the current error covariance. In fact, it can be shown that any cost function of the form

$$J = E[\epsilon^T A \epsilon]$$

can be uniquely minimized with respect to K only if A is positive semi-definite (see Geib, 1975 for example). Of course, the trace of P is only a special case when $A = I$. One could

also choose the positive definite matrix $A = \phi^T(t_p) \phi(t_p)$ as we have and minimize the prediction error. For the case of $n = 2$, we are interested in

$$\sigma_1(t + t_p | t) = [P_{11}(t|t) + 2t_p P_{12}(t|t) + t_p^2 P_{22}(t|t) + t_p^2 P_{13}(t|t) + t_p^3 P_{23}(t|t) + t_p^4 P_{33}(t|t)/4]^{1/2} \quad (3.23)$$

A prediction time of $t_p = 10$ seconds is used as a standard of comparison for all the one-dimensional results in this study. This corresponds to a nominal projectile time-of-flight for a Navy five-inch gun system engaging air targets. Notice also in Equation (3.23) the factors that multiply the rate errors. It can clearly be seen that velocity and acceleration estimation errors are the principal cause of prediction noise.

Before discussing the convergence results, it might be helpful to consider a transformation of the polynomial filter, suggested by Morrison (1969), that is useful in understanding and interpreting these results. A transformation of the state vector can be used to absorb the Δt dependence in the error covariance equations. This is done by defining an alternate state vector

$$x^*(t) = [x \quad (\Delta t) dx/dt \quad (\Delta t^2/2) d^2x/dt^2 \quad \dots \quad (\Delta t^n/n!) d^n x/dt^n]^T \quad (3.23)$$

The equivalent transition matrix is

$$\phi_{ij}^* = (j-1)! / [(i-1)!(j-i)!] \quad (3.24)$$

which, we find, is a matrix of constants independent of Δt . For the example, $n = 2$, we find

$$\phi^* = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

If the Δt dependence is removed from the initial error covariance, the error covariance and gains are then a function only of the data point k and the order n . In subsequent sections, we will show that the frequency content of target motion is rather low so that a high sampling rate is not actually necessary to satisfy the Shannon sampling theorem. In fact, the principal advantage of a high data rate is to reduce the effective measurement error level. In the section on prefiltering, it will be shown that data compression techniques can be used to achieve effective high data rates without actually cycling the Kalman filter at such rates.

In order to consider the real convergence criterion, the extrapolated position error standard deviation, it is more straightforward to use the original unscaled equations (without Morrison's transformation). In Figure 3.1, we have plotted the time history of the standard deviation of the predicted position error ($t_p = 10$ seconds), normalized by σ (the measurement error level), for each filter order and data rate pair (n, k) under consideration. Such a graph can be used to determine optimal settling time for any particular situation. For example, suppose we decide the filter is "settled" enough to fire the gun when the predicted target position is accurate to within 25 yards. If we can track the target with a $\sigma = 5$ yards, then using the (2, 16) filter, we could start firing 8 seconds after the initiation of target track. Basically, we see from this graph that the lower the order and the higher the data rate, the faster convergence will occur. We see that the third-order filter converges painfully slowly at these data rates and is probably not suitable for gunfire control purposes due to this factor and the heavy real-time computational burden of this filter at high data rates. The constant-velocity filter, on the other hand, converges very rapidly, predicting better than the tracking measurement error in less than 10 seconds, with a data rate of only 4 Hertz.

Of course, settling time of these optimal filters is not the only consideration in making a choice. We know that targets do accelerate and that the constant-velocity filter, without an estimate of acceleration, will do quite poorly in predicting future target position. The error in that situation is of a bias type rather than the uncorrelated error considered here. In Figure 3.2, we have plotted for reference purposes the complete error convergence history of all the state elements (position, velocity, acceleration, and 10-second predicted position) for the constant acceleration ($n = 2$) filter. With the dashed lines, we have also plotted the corresponding actual error that would occur if we attempted to filter the "real-world" constant acceleration model by assuming a constant velocity ($n = 1$) filter model. Large bias errors in all state elements are observed to occur in the actual error elements when using an $n = 1$ filter in an $n = 2$ environment. The prediction error, which initially starts to get smaller, quickly reverses itself and satisfactory prediction error is never achieved. The $n = 1$ filter is really not a candidate if we are going to attempt to deal with maneuvering targets. That leaves the $n = 2$ filter which we would choose to implement if a polynomial target model were selected. We would also choose the $k = 16$ Hertz (or faster) since it converges fastest. Convergence time is not only important initially, but, as we shall see in a later section on adaptation, it may be necessary for the filter to reconverge when the adaptive features in the filter decide it necessary. Filter responsiveness in such a situation dictates greater significance to the problem of realizing a short settling time.

In conclusion, then, we would choose the second-order filter over the third-order (ruled out due to settling time) and over the first-order (because we must track an accelerating target in an effective manner). We also conclude that, given uncorrelated measurement noise (satisfying the requirement for the derivation of this filter), we want to process data at the highest rate possible. The final determination of this rate will be based on available

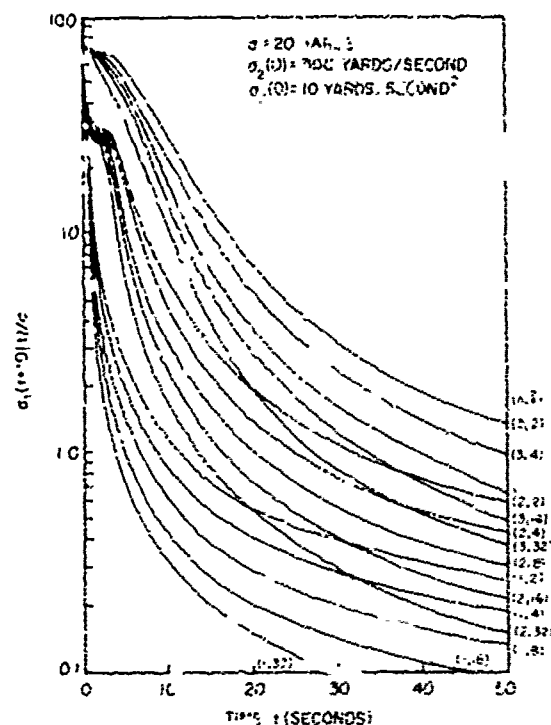


Figure 3.1. Derivative Polynomial Filters with Uncorrelated One-Pass Initialization Normalized Standard Deviation of Predicted (10 sec) Position Error

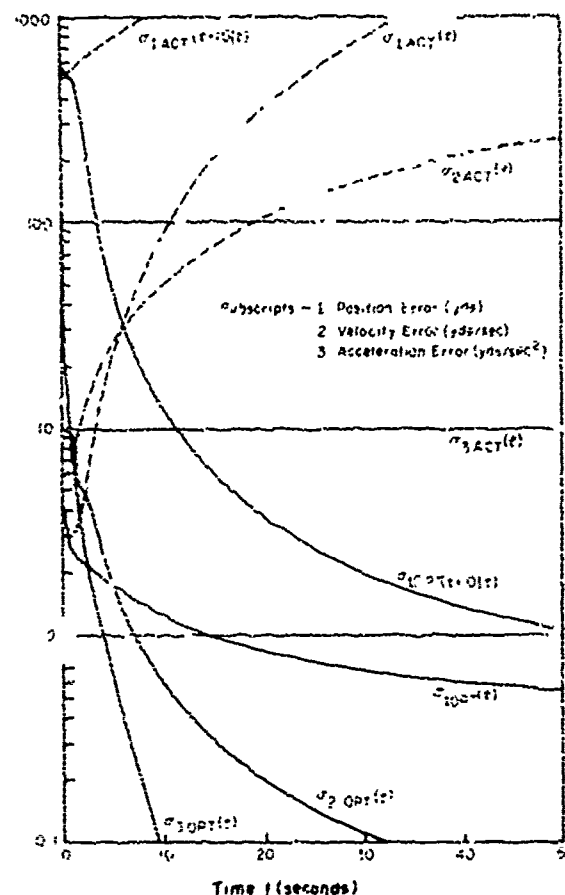


Figure 3.2. Optimal Convergence of $n = 2$ Filter and Actual Error Using $n = 1$ Filter

computation time in the real-time filter implementation and on the ability to determine and remove the actual correlated portion of the measurement noise. This last aspect of the problem will be discussed in a later section. We will proceed on the basis of $k = 16$ Hertz unless otherwise noted. For reference purposes, a summary of the filter equations for $n = 2$ can be found in Table 3.1.

Another possibility, which the author would like to explore in the future, is that of polynomial filter cascading. In this approach, one could conceivably obtain the fast convergence properties of the low-order filter and the modeling improvement of the high-order filters. The procedure might be implemented in the following manner. A constant velocity

filter ($n = 1$), with limited memory to prevent divergence (to be discussed later), would first be implemented. The output state vector of this filter would then provide the measurements and measurement error covariance for a constant-acceleration ($n = 2$) filter (also with specified memory). It might even be possible to use another cascade and feed a jerque ($n = 3$) filter. A logic section would then operate on the residuals of each filter and pick the one that is operating the best. The possibilities are certainly intriguing and should be explored if possible.

Table 3.1 Second-Order Filter Equations: A Summary

EXTRAPOLATION

$$\hat{x}_1(k|k-1) = \hat{x}_1(k-1|k-1) + \Delta t \hat{x}_2(k-1|k-1) + \frac{\Delta t^2}{2} \hat{x}_3(k-1|k-1) \quad (1)$$

$$\hat{x}_2(k|k-1) = \hat{x}_2(k-1|k-1) + \Delta t \hat{x}_3(k-1|k-1) \quad (2)$$

$$\hat{x}_3(k|k-1) = \hat{x}_3(k-1|k-1) \quad (3)$$

$$\begin{aligned} P_{11}(k|k-1) = & P_{11}(k-1|k-1) + 2\Delta t P_{12}(k-1|k-1) + \Delta t^2 P_{13}(k-1|k-1) \\ & + \Delta t^2 P_{22}(k-1|k-1) + \Delta t^3 P_{23}(k-1|k-1) + \frac{\Delta t^4}{4} P_{33}(k-1|k-1) \\ & + Q_{11}(k-1) \end{aligned} \quad (4)$$

$$\begin{aligned} P_{12}(k|k-1) = & P_{12}(k-1|k-1) + \Delta t P_{13}(k-1|k-1) + \Delta t P_{22}(k-1|k-1) \\ & + \frac{3}{2} \Delta t^2 P_{23}(k-1|k-1) + \frac{\Delta t^3}{2} P_{33}(k-1|k-1) + Q_{12}(k-1) \end{aligned} \quad (5)$$

$$\begin{aligned} P_{13}(k|k-1) = & P_{13}(k-1|k-1) + \Delta t P_{23}(k-1|k-1) + \frac{\Delta t^2}{2} P_{33}(k-1|k-1) \\ & + Q_{13}(k-1) \end{aligned} \quad (6)$$

$$\begin{aligned} P_{22}(k|k-1) = & P_{22}(k-1|k-1) + 2\Delta t P_{23}(k-1|k-1) + \Delta t^2 P_{33}(k-1|k-1) \\ & + Q_{22}(k-1) \end{aligned} \quad (7)$$

$$P_{23}(k|k-1) = P_{23}(k-1|k-1) + \Delta t P_{33}(k-1|k-1) + Q_{23}(k-1) \quad (8)$$

$$P_{33}(k|k-1) = P_{33}(k-1|k-1) + Q_{33}(k-1) \quad (9)$$

Table 3.1. Second-Order Filter Equations: A Summary -(Continued)

GAINS

$$K_1(k) = P_{11}(k|k-1)/[P_{11}(k|k-1) + \sigma^2(k)] \quad (10)$$

$$K_2(k) = P_{12}(k|k-1)/[P_{11}(k|k-1) + \sigma^2(k)] \quad (11)$$

$$K_3(k) = P_{13}(k|k-1)/[P_{11}(k|k-1) + \sigma^2(k)] \quad (12)$$

RESIDUAL

$$v(k|k-1) = z(k) - \hat{y}_1(k|k-1) \quad (13)$$

UPDATE

$$\hat{x}_1(k|k) = \hat{x}_1(k|k-1) + K_1(k) v(k|k-1) \quad (14)$$

$$\hat{x}_2(k|k) = \hat{x}_2(k|k-1) + K_2(k) v(k|k-1) \quad (15)$$

$$\hat{x}_3(k|k) = \hat{x}_3(k|k-1) + K_3(k) v(k|k-1) \quad (16)$$

$$P_{11}(k|k) = [1 - K_1(k)] P_{11}(k|k-1) \quad (17)$$

$$P_{12}(k|k) = [1 - K_1(k)] P_{12}(k|k-1) \quad (18)$$

$$P_{13}(k|k) = [1 - K_1(k)] P_{13}(k|k-1) \quad (19)$$

$$P_{22}(k|k) = P_{22}(k|k-1) - K_2(k) P_{12}(k|k-1) \quad (20)$$

$$P_{23}(k|k) = P_{23}(k|k-1) - K_2(k) P_{13}(k|k-1) \quad (21)$$

$$P_{33}(k|k) = P_{33}(k|k-1) - K_3(k) P_{13}(k|k-1) \quad (22)$$

INITIALIZATION

$$\hat{x}_1(0|0) = z(0) \quad (23)$$

$$\hat{x}_2(0|0) = \hat{x}_3(0|0) = 0 \quad (24)$$

Table 3.1. Second-Order Filter Equations: A Summary--(Continued)

INITIALIZATION--(Continued)

$$P_{11}(0|0) = \sigma^2(0) \quad (25)$$

$$P_{12}(0|0) = 0. \quad (26)$$

$$P_{13}(0|0) = 0. \quad (27)$$

$$P_{22}(0|0) = \sigma_2^2(0) \quad (28)$$

$$P_{23}(0|0) = 0. \quad (29)$$

$$P_{33}(0|0) = \sigma_3^2(0) \quad (30)$$

PREDICTION

$$\hat{x}_1(t+t_p|t) = \hat{x}_1(t|t) + \hat{x}_2(t|t)t_p + \hat{x}_3(t|t)t_p^2/2 \quad (31)$$

RECOMMENDED VALUES OF PARAMETERS

$$\dot{k} = 16 \text{ Hertz} \quad (32)$$

FOR AIR TARGETS

$$\sigma_2(0) = 300 \text{ yards/second} \quad (33)$$

$$\sigma_3(0) = 10 \text{ yards/second}^2 \quad (34)$$

FOR SURFACE (S. IP) TARGETS

$$\sigma_2(0) = 14 \text{ yards/second} \quad (35)$$

$$\sigma_3(0) = 2 \text{ yard/second}^2 \quad (36)$$

IV. THE RANDOM-ACCELERATION TARGET MODEL

The derivative polynomial target model of Section III, while possessing several advantages, undoubtedly suffers as far as being a realistic representation of the scenario of actual targets over a period of more than a few seconds. Actual targets would rarely traverse a simple *predictable* path while in the range of engagement by our GFCs. Instead, such targets would be *maneuvering*. These maneuvers could take the form of a deterministic path if the target has a particular goal in mind, such as effecting a collision with the ship (in the case of an anti-ship missile) or the release of ordnance (in the case of a bomber). The scenario would also have to include nondeterministic (or random) maneuvers in such situations when the target is maneuvering merely to evade interception by the ship's weapons or when irregular winds and atmospheric turbulence would act on the target. Anti-ship missiles would be subject to the latter effects and also may respond to the wander of its own radar around the target ship. In any case, the probability of a target following a constant acceleration (or any polynomial) course for any length of time is, unfortunately, not what we might prefer. With this idea in mind, perhaps the problem can be approached from a different point of view. Rather than attempt to model the target with a deterministic trajectory such as a polynomial, let us consider a case where we acknowledge the fact that the target is very likely to be maneuvering in some unknown manner and where we assume that each dimension of such trajectories appear (to the FCS) to be a random variable. The following model was developed by Singer (1970) and has already been selected for implementation in several other fire control and tracking systems.

The target acceleration is assumed to be an autocorrelated (or serially correlated) random variable with known statistics. Specifically, Singer chose to model the acceleration as a first-order Markov process with the differential equation

$$\frac{d}{dt} \left(\frac{d^2 x(t)}{dt^2} \right) = - \frac{1}{\tau_m} \left(\frac{d^2 x(t)}{dt^2} \right) + w(t) \quad (4.1)$$

driven by a white noise input of variance

$$\sigma_w^2(T) = \frac{2g_m^2}{\tau_m} \delta(T) \quad (4.2)$$

The acceleration sequence generated is a form of process known as a random walk. An example of such a process generated on the computer with a correlated random number generator CORNUM (See Appendix A) is shown in Figure 4.1. Notice that the corresponding position (double integral of \ddot{x}) is also plotted. The autocorrelation function for the acceleration is the exponential

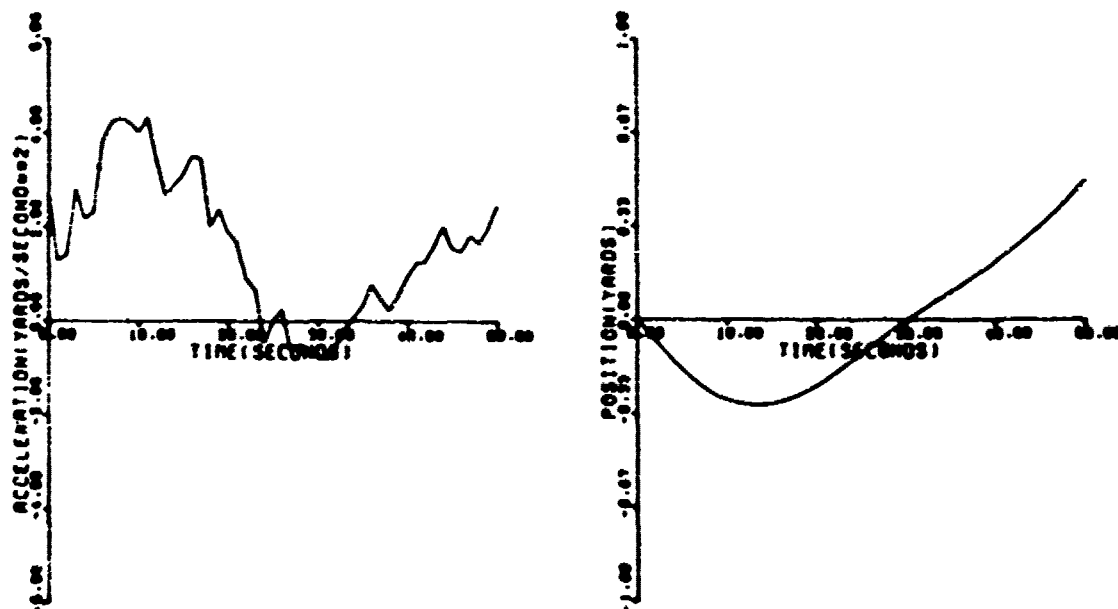


Figure 4.1. Example: Random Acceleration Model
 $(\tau_m = 20 \text{ seconds}, \sigma_m = 2 \text{ yards/second}^2)$

$$\begin{aligned}\Omega_{\ddot{x}}(T) &= E[\ddot{x}(t) \ddot{x}(t+T)] \\ &= \sigma_m^2 \exp(-|T|/\tau_m)\end{aligned}\tag{4.3}$$

We therefore require only two parameters (σ_m and τ_m or ω_m) to characterize the target maneuverability. The parameter σ_m can be (loosely) thought of as the root-mean-square level of target acceleration and τ_m as the characteristic maneuver time (approximately 0.8 times the mean time between acceleration zeros) or the inverse of maneuver frequency ω_m . Notice that since σ_m is essentially the rms level of acceleration for one dimension, the actual peak total (three-dimensional) acceleration might be several (perhaps three to four) times larger than σ_m . It is felt that this model is much better suited to our purposes than the polynomial models because it directly relates the filter parameters to the kinematics of the target scenario. We have also found that this model performs better in both filtering and prediction than either the $n = 1$ or $n = 2$ polynomial models. It should be emphasized that we do not mean to imply that one cannot find particular trajectories that might poorly match this model. We do expect actual trajectory to be "contained" statistically in the random acceleration model and thus be a "reasonable" sample from our assumed population. The author feels that the random acceleration model, while far from perfect, is the best model yet developed for general tracking filter application. The model can be written in state space notation by defining the state vector

$$\underline{x}(t) = [x(t) \ dx(t)/dt \ d^2x(t)/dt^2]^T \quad (4.4)$$

and

$$d\underline{x}(t)/dt = F(t) \underline{x}(t) + \underline{w}(t) \quad (4.5)$$

where

$$F(t) = F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_m \end{bmatrix} \quad (4.6)$$

and

$$\underline{w}(t) = [0 \quad 0 \quad w(t)]^T \quad (4.7)$$

By simple integration of Equation (4.6), we find that the transition matrix can be written as

$$\phi(k, k-1) = \phi = \begin{bmatrix} 1 & \Delta t & \alpha(\Delta t) \\ 0 & 1 & \beta(\Delta t) \\ 0 & 0 & \gamma(\Delta t) \end{bmatrix} \quad (4.8)$$

where

$$\gamma(\Delta t) = \exp(-\Delta t/\tau_m) \quad (4.9a)$$

$$\beta(\Delta t) = \tau_m [1 - \gamma(\Delta t)] \quad (4.9b)$$

$$\alpha(\Delta t) = \tau_m^2 [\gamma(\Delta t) + \Delta t/\tau_m - 1] \quad (4.9c)$$

The argument of α , β and $\gamma - \Delta t -$ will hereafter be suppressed unless specified for a particular purpose. The resemblance of this model to the second-order polynomial model can be seen immediately if we consider the limiting case of $\tau_m \gg \Delta t$. The transition matrix becomes:

$$\phi \cong \phi_2 = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

i.e. identical to the second-order polynomial model. On the other hand, if τ_m vanishes (or $\tau_m \ll \Delta t$), we find

$$\phi \cong \phi_1 = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.11)$$

This transition matrix corresponds to the first-order polynomial model. We would therefore expect, finite values of τ_m (between zero and infinity) to yield a model that exhibits convergence properties between those of the first- and second-order polynomial filters. Indeed, this turns out to be the case. In Figure 4.2, the extrapolated position error standard deviation is again plotted for various values of τ_m . The cases $\tau_m = 0$ and $\tau_m \rightarrow \infty$ corresponds exactly to the first- and second-order polynomial cases. All cases were run with a data rate of 16 Hertz, a value of $\sigma_m = 0$ (limiting case of σ_m small) and with one-pass initialization. We find that, just as expected, as τ_m gets larger, the convergence properties move toward the second-order polynomial case.

The process noise matrix, Q , is developed in the appendix of Singer's paper. This development is rather lengthy and will not be reproduced here. The final results can be written

$$Q_{11} = \tau_m^4 \sigma_m^2 \{ 1 + 1(\Delta t/\tau_m) - 2(\Delta t/\tau_m)^2 + 2(\Delta t/\tau_m)^3/3 - 4(\Delta t/\tau_m)\gamma - \gamma^2 \} \quad (4.12a)$$

$$Q_{12} = \tau_m^3 \sigma_m^2 \{ 1 - \gamma^2 - 2\gamma + 2(\Delta t/\tau_m)\gamma - 2(\Delta t/\tau_m) + (\Delta t/\tau_m)^2 \} \quad (4.12b)$$

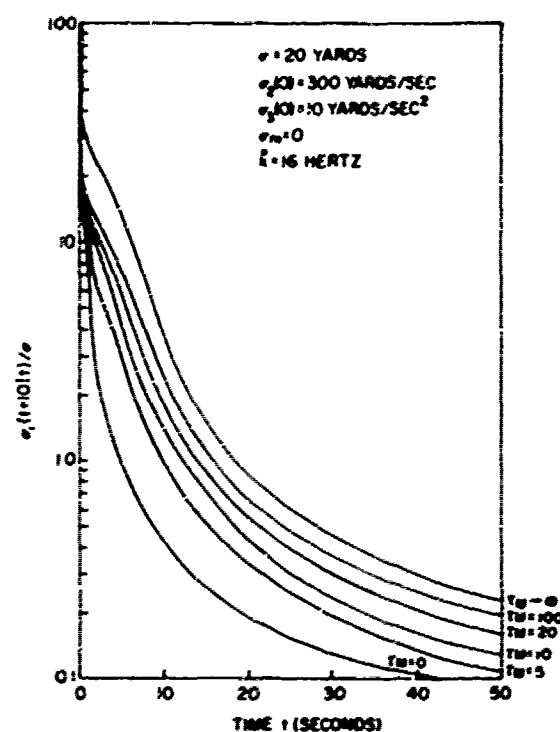
$$Q_{13} = \tau_m^2 \sigma_m^2 \{ 1 - \gamma^2 - 2(\Delta t/\tau_m)\gamma \} \quad (4.12c)$$

$$Q_{22} = \tau_m^2 \sigma_m^2 \{ 4\gamma - 3 - \gamma^2 + 2(\Delta t/\tau_m) \} \quad (4.12d)$$

$$Q_{23} = \tau_m \sigma_m^2 (1 + \gamma^2 - 2\gamma) \quad (4.12e)$$

$$Q_{33} = \sigma_m^2 (1 - \gamma^2) \quad (4.12f)$$

Figure 4.2. Random Acceleration Filter with Various Values of normalized Standard Deviation of Predicted (10 sec) Position Error



(Notice that Q is the only place where σ_M appears in the filter.) Again for the limiting case of $\tau_M \gg \Delta t$, this matrix can be approximated as

$$Q = 2\Delta t \sigma_M^2 / \tau_M \begin{bmatrix} \Delta t^4/20 & \Delta t^3/8 & \Delta t^2/6 \\ \Delta t^3/8 & \Delta t^2/6 & \Delta t/2 \\ \Delta t^2/6 & \Delta t/2 & 1 \end{bmatrix} \quad (4.13)$$

Since we will always operate with parameters Δt and τ_M in this latter range, we will use this approximate form of Q for our work. Simulations show, that for values of the parameters Δt and τ_M in the ranges considered in this report, there is no difference between the use of Equation (4.13) instead of Equation (4.12). We must use the exact expression for Q in the long-time extrapolated covariance however. Notice in Equation (4.13) that a decrease in τ_M or an increase in frequency influences Q in the same way as an increase in σ_M^2 . The extrapolated position variance equation for this case is

$$\begin{aligned} P_{11}(t + t_p/t) = & P_{11}(t/t) + 2 t_p P_{12}(t/t) + 2 \alpha_p P_{13}(t/t) \\ & + t_p^2 P_{22}(t/t) + 2 \alpha_p t_p P_{23}(t/t) + \alpha_p^2 P_{33}(t/t) \\ & + Q_{11p} \end{aligned} \quad (4.14)$$

where

$$\alpha_p = \tau_M^2 [\exp(-t_p/\tau_M) + t_p/\tau_M - 1] \quad (4.15)$$

and

$$\begin{aligned} Q_{11p} = & \tau_M^4 \sigma_M^2 [1 + 2(t_p/\tau_M) - 2(t_p/\tau_M)^2 \\ & + 2(t_p/\tau_M)^3/3 - 4(t_p/\tau_M) \exp(-t_p/\tau_M) \\ & - \exp(-2t_p/\tau_M)] \end{aligned} \quad (4.16)$$

The choice of a good value of τ_M is a function primarily of the target scenario for the GFCS. The autocorrelation was studied for several typical profiles of older anti-ship missiles, and a value of approximately $\tau_M = 20$ seconds appeared to consistently be yielded. Singer (1970) recommends $\tau_M = 20$ seconds also for manned maneuvering targets exercising evasive maneuvers. The same value has also been found independently by other people who have studied the problem. No information on maneuvering surface targets has yet been analyzed. Other values can, of course, be chosen for surface targets or to reduce settling time as may be required. It should be emphasized that this rather low *observed* maneuver frequency, corresponding to $\tau_M = 20$ seconds, does not necessarily imply that this is the highest frequency that a particular target might be capable of sustaining. On the contrary, most air targets can maneuver much more rapidly if desired. Rather, a low maneuver frequency is probably typical of air targets maneuvering to achieve a particular goal such as intercepting ownship or exercising their own fire control in order to release ordnance at ownship. We consider values of τ_M anywhere in the range of 3 to 20 seconds as realistic.

The selection of a value of σ_M is more difficult than that for τ_M . The acceleration autocorrelation study of the target scenario demonstrates a very wide range of values for σ_M . For example, some targets might achieve an rms maneuver level close to one G while others are essentially nonmaneuvering (actually maneuvering at a very low level due to atmospheric effects). The effect, however, of σ_M on optimal filter performance is quite dramatic. For example, in Figure 4.3, the steady-state value of the normalized 10-second prediction error is plotted as a function of σ_M for $\tau_M = 20$ seconds. For a value of $\sigma_M = 10$ yards/second² (at the upper limit close to one G), the normalized prediction error is 15.96, clearly a very poor value for GFCS applications with conventional projectiles. The reason such a large prediction error occurs is due to the value of the current error covariance which remains large due to the process noise.

Figure 4.3 also plots the current acceleration estimation error level which we see is quite sensitive to σ_M . Since the prediction error equation (4.14) magnifies the acceleration error, we find the large prediction errors occurring for large σ_M . Obviously, the parameter σ_M has a significant effect on the filter bandwidth and, since it has such a wide range of

values, must be carefully chosen to achieve proper filter operation. The next section on adaption will deal in more detail with the specification of σ_M and τ_M . For reference purposes, the convergence of the normalized prediction error is plotted in Figure 4.4 for a range of values of σ_M and τ_M . Notice that the steady state values for smaller τ_M (higher ω_m) are not necessarily larger for prediction since smaller τ_M suppresses prediction noise at the same time it increases filter noise.

It is also interesting to determine the sensitivity to incorrect assumed values for the random acceleration parameters. To do this, we first determine a set of parameters σ_{Mf} and τ_{mf} for the filter (subscript f) to assume. We then allow one of the parameters to vary and calculate the actual and optimal covariance with the equations of Section II. We define a figure of merit

$$\theta(t_p, T) = \frac{1}{KT} \sum_{k=1}^{K=KT} \sigma_{opt}(t_k + t_p/t_k) / \sigma_{ACT}(t_k + t_p/t_k) \quad (4.17)$$

as the integrated (average) ratio of the optimal to the actual predicted position error standard deviation. Since σ_{ACT} is always greater than or equal to σ_{opt} , θ is always less than unity and equal to one only for the optimal case. We might therefore think of θ as the degree to which the suboptimal filter matches the optimal performance. Using prediction time $t_p = 10$ seconds and total run time $T = 50$ seconds as usual, the sensitivity results are plotted in Figure 4.5 for three different filter bandwidths wide (w), medium (M) and narrow (N). In Figure 4.5(a), we find it takes an error of approximately an order of magnitude in assumed target maneuver level to produce a 50 percent optimality level i.e. when the actual error is twice what it could be. It is also slightly preferable if we must be in error to underestimate than to overestimate the maneuver level σ_M . In Figure 4.5(b), we find that prediction error levels are practically insensitive to the choice of the maneuver time constant τ_M . The reason for this behavior is probably related to the filter-predictor offsetting effects mentioned in the last paragraph. Notice in both figures, also, that there is very little (if any) discernible difference in sensitivity over the significant range of filter bandwidths.

The author has therefore chosen the random-acceleration target model over the polynomial model primarily because the random-acceleration model directly relates the target's kinematics to the filter transition and process noise matrices and therefore to the filter bandwidth. All the extrapolation equations will change from those presented for the polynomial case. The update equations do not change. Specifying the one-pass initialization completes the description of the new filter. For reference purposes, a summary of these filter equations appears in Table 4.1.

Before continuing to the next section, it should be mentioned that other nondeterministic target models are under consideration for the GFCS tracking application. Brown and Price (1974), for example, tried a higher-order analogy of the model in this section.

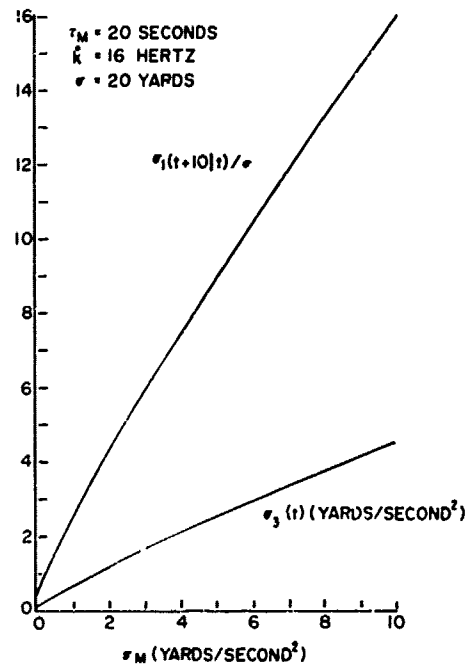


Figure 4.3. Random Acceleration Filter Steady State Values as Function of σ_M

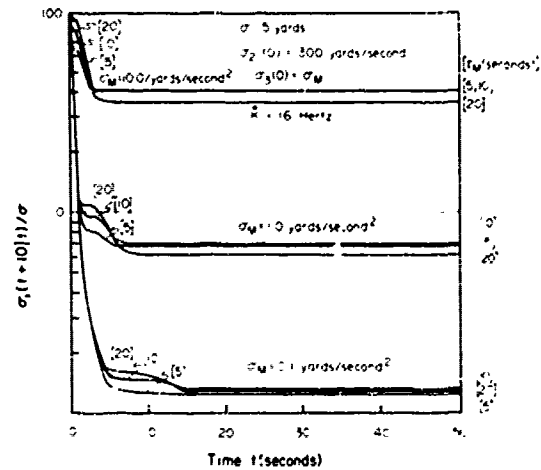


Figure 4.4. Random Acceleration Filter with Various Values of σ_M and τ_M - Normalized Standard Deviation at Predicted (10 second) Position Error

ie, acceleration *rate* is an exponentially correlated random variable, but found that it did not work as well as the model discussed here. Moose (1972) discusses a very interesting model, known as a semi-Markov process, which he applied to the case of a maneuvering submarine. The author would like to investigate the applicability of this model to the air-target tracking problem.

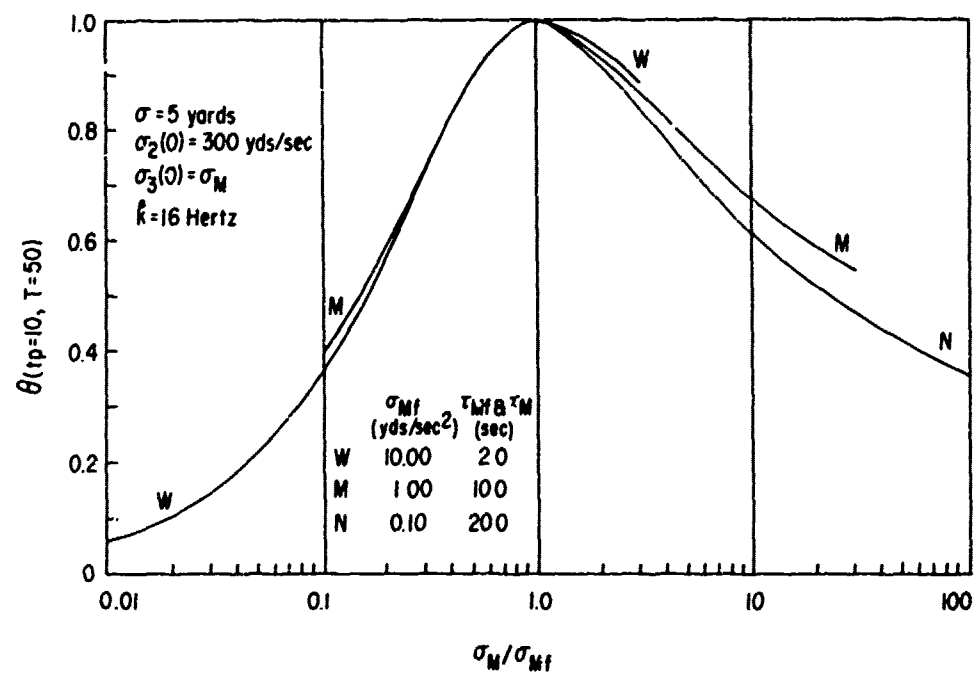
Table 4.1 Random Acceleration Filter Equations. A Summary

EXTRAPOLATION

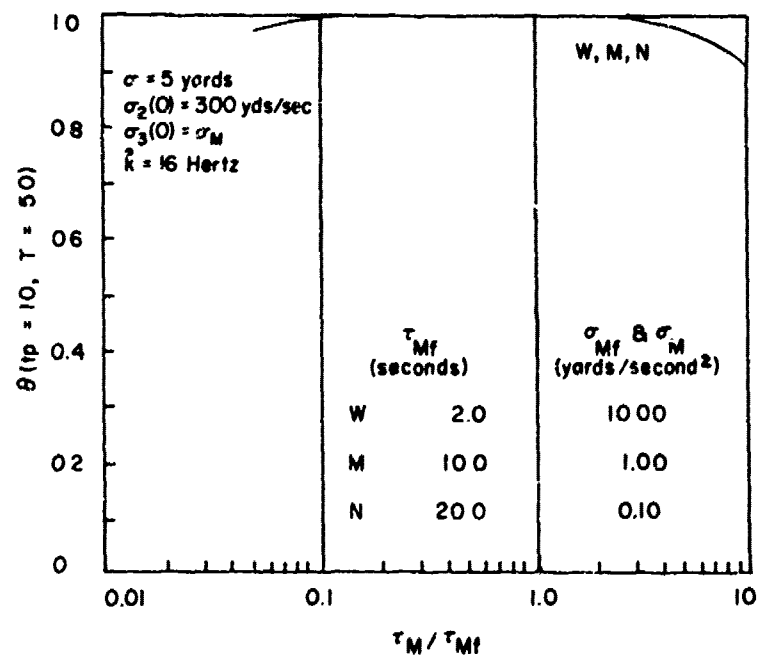
$$\hat{x}_1(k/k-1) = \hat{x}_1(k-1/k-1) + \Delta t \hat{x}_2(k-1/k-1) + \alpha \hat{x}_3(k-1/k-1) \quad (1)$$

$$\hat{x}_2(k/k-1) = \hat{x}_2(k-1/k-1) + \beta \hat{x}_3(k-1/k-1) \quad (2)$$

$$\hat{x}_3(k/k-1) = \gamma \hat{x}_3(k-1/k-1) \quad (3)$$



(a) Sensitivity to an Incorrect Acceleration Level



(b) Sensitivity to an Incorrect Acceleration Time Constant

Figure 4.5

Table 4.1. Random Acceleration Filter Equations: A Summary--(Continued)

$$\begin{aligned} P_{11}(k/k-1) = & P_{11}(k-1/k-1) + 2\Delta t P_{12}(k-1/k-1) + 2\alpha P_{13}(k-1/k-1) \\ & + \Delta t^2 P_{22}(k-1/k-1) + 2\alpha \Delta t P_{23}(k-1/k-1) \\ & + \alpha^2 P_{33}(k-1/k-1) + Q_{11}(k-1) \end{aligned} \quad (4)$$

$$\begin{aligned} P_{12}(k/k-1) = & P_{12}(k-1/k-1) + \beta P_{13}(k-1/k-1) + \Delta t P_{22}(k-1/k-1) \\ & + (\alpha + \beta \Delta t) P_{23}(k-1/k-1) + 2\beta P_{33}(k-1/k-1) + Q_{12}(k-1) \end{aligned} \quad (5)$$

$$\begin{aligned} P_{13}(k/k-1) = & \gamma P_{13}(k-1/k-1) + \gamma \Delta t P_{23}(k-1/k-1) + \alpha \gamma P_{33}(k-1/k-1) \\ & + Q_{13}(k-1) \end{aligned} \quad (6)$$

$$\begin{aligned} P_{22}(k/k-1) = & P_{22}(k-1/k-1) + 2\beta P_{23}(k-1/k-1) + \beta^2 P_{33}(k-1/k-1) \\ & + Q_{22}(k-1) \end{aligned} \quad (7)$$

$$P_{23}(k/k-1) = \gamma P_{23}(k-1/k-1) + \beta \gamma P_{33}(k-1/k-1) + Q_{23}(k-1) \quad (8)$$

$$P_{33}(k/k-1) = \gamma^2 P_{33}(k-1/k-1) + Q_{33}(k-1) \quad (9)$$

GAINS

$$K_1(k) = P_{11}(k/k-1) / [P_{11}(k/k-1) + \sigma^2(k)] \quad (10)$$

$$K_2(k) = P_{12}(k/k-1) / [P_{11}(k/k-1) + \sigma^2(k)] \quad (11)$$

$$K_3(k) = P_{13}(k/k-1) / [P_{11}(k/k-1) + \sigma^2(k)] \quad (12)$$

RESIDUAL

$$v(k/k-1) = z(k) - \hat{x}_1(k/k-1) \quad (13)$$

Table 4.1. Random Acceleration Filter Equations: A Summary--(Continued)

UPDATE

$$\hat{x}_1(k/k) = \hat{x}_1(k/k-1) + K_1(k)v(k/k-1) \quad (14)$$

$$\hat{x}_2(k/k) = \hat{x}_2(k/k-1) + K_2(k)v(k/k-1) \quad (15)$$

$$\hat{x}_3(k/k) = \hat{x}_3(k/k-1) + K_3(k)v(k/k-1) \quad (16)$$

$$P_{11}(k/k) = [1 - K_1(k)]P_{11}(k/k-1) \quad (17)$$

$$P_{12}(k/k) = [1 - K_1(k)]P_{12}(k/k-1) \quad (18)$$

$$P_{13}(k/k) = [1 - K_1(k)]P_{13}(k/k-1) \quad (19)$$

$$P_{22}(k/k) = P_{22}(k/k-1) - K_2(k)P_{12}(k/k-1) \quad (20)$$

$$P_{23}(k/k) = P_{23}(k/k-1) - K_2(k)P_{13}(k/k-1) \quad (21)$$

$$P_{33}(k/k) = P_{33}(k/k-1) - K_3(k)P_{13}(k/k-1) \quad (22)$$

INITIALIZATION

$$\hat{x}_1(0/0) = z(0) \quad (23)$$

$$\hat{x}_2(0/0) = \hat{x}_3(0/0) = 0 \quad (24)$$

$$P_{11}(0/0) = \sigma^2(0) \quad (25)$$

$$P_{12}(0/0) = 0. \quad (26)$$

$$P_{13}(0/0) = 0. \quad (27)$$

$$P_{22}(0/0) = \sigma_2^2(0) \quad (28)$$

$$P_{23}(0/0) = 0. \quad (29)$$

$$P_{33}(0/0) = \sigma_3^2(0) \quad (30)$$

Table 4.1. Random Acceleration Filter Equations: A Summary--(Continued)

CONSTANTS

$$\gamma = \exp(-\Delta t/\tau_M) \quad (31)$$

$$\beta = \tau_M(1 - \gamma) \quad (32)$$

$$\alpha = \tau_M^2(\gamma + \Delta t/\tau_M - 1) \quad (33)$$

PROCESS NOISE (CONSTANT)

$$Q_{33} = 2\Delta t\sigma_M^2/\tau_M \quad (34)$$

$$Q_{23} = \Delta t Q_{33}/2 \quad (35)$$

$$Q_{22} = 2\Delta t Q_{23}/3 \quad (36)$$

$$Q_{13} = Q_{22}/2 \quad (37)$$

$$Q_{12} = 3\Delta t Q_{13}/4 \quad (38)$$

$$Q_{11} = 2\Delta t Q_{12}/5 \quad (39)$$

PREDICTION

$$\begin{aligned} \hat{x}_1(t + t_p/t) = & \hat{x}_1(t/t) + \hat{x}_2(t/t)\dot{t}_p + \hat{x}_3(t/t)\tau_M^2[\exp(-t_p/\tau_M) \\ & + t_p/\tau_M - 1] \end{aligned} \quad (40)$$

RECOMMENDED VALUES OF PARAMETERS

$$\dot{k} = 16 \text{ Hertz} \quad (41)$$

$$\sigma_M = 0.1\text{--}10.0 \text{ yards/second}^2 \quad (42)$$

$$\tau_M = 3\text{--}20 \text{ seconds} \quad (43)$$

Table 4.1. Random Acceleration Filter Equations: A Summary--(Continued)

FOR AIR TARGETS

$$\sigma_2(0) = 300 \text{ yards/second} \quad (44)$$

$$\sigma_3(0) = 10 \text{ yards/second}^2 \quad (45)$$

FOR SURFACE TARGETS

$$\sigma_2(0) = 14 \text{ yards/second} \quad (46)$$

$$\sigma_3(0) = 2 \text{ yard/second}^2 \quad (47)$$

V. ADAPTATION

The Kalman filter formulation, presented in the previous sections, assumes complete knowledge of the linear dynamic model and the process noise covariance. In a general tracking filter application, such as the gunfire control problem, the particular strategy being exercised by the target is unknown. The form of the state vector and its propagation characteristics is assumed and may or may not adequately represent the true target motion over long periods of time. (We always expect, however, the dynamics model to be a good approximation of target motion over short periods of time.) Such a situation is usually referred to a "suboptimal modeling" in the sense that no attempt is made to fully model the target dynamics. The utilization of a suboptimal model often leads to large estimation errors—a condition known as filter divergence. When divergence occurs, an inconsistency between the error covariance calculated by the filter and the actual error covariance occurs. Examples of such divergence problems will be shown shortly.

An adaptive filter is basically a method of adjusting parameters in order to effect a more realistic match between the calculated and actual filter error covariances. The purpose of such a technique is to reduce and bound the actual error covariance when modeling errors become large enough to seriously affect the performance. We will find that, contrary to the linear Kalman filter techniques described thus far, the calculated error covariance must become a function of the actual data through a coupling of the filter parameters with the target motion. We will also find that the performance of a particular adaptive filter is a function of the procedure used to detect divergence and of the method used to modify the filter when divergence is detected. After the following examples, techniques of detecting divergence and two forms of adaptive filters will be presented.

A. DIVERGENCE AND THE BANDWIDTH TRADEOFF

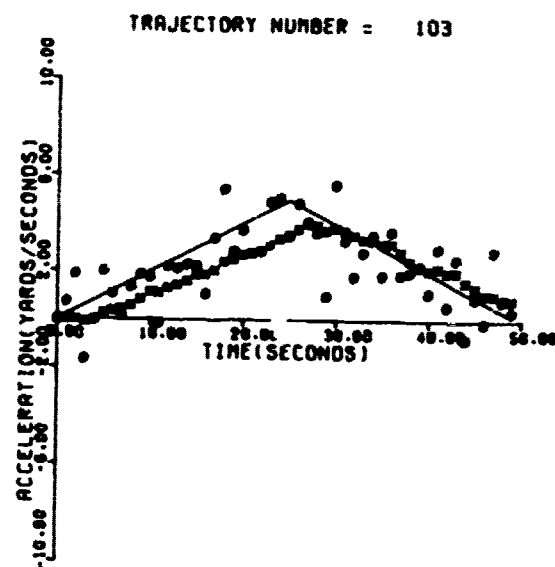
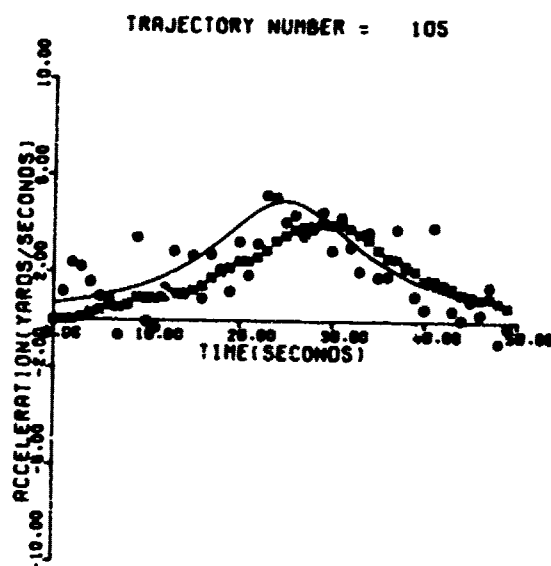
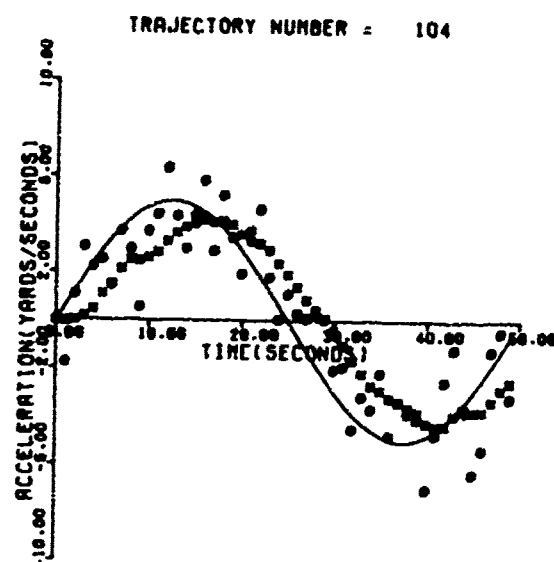
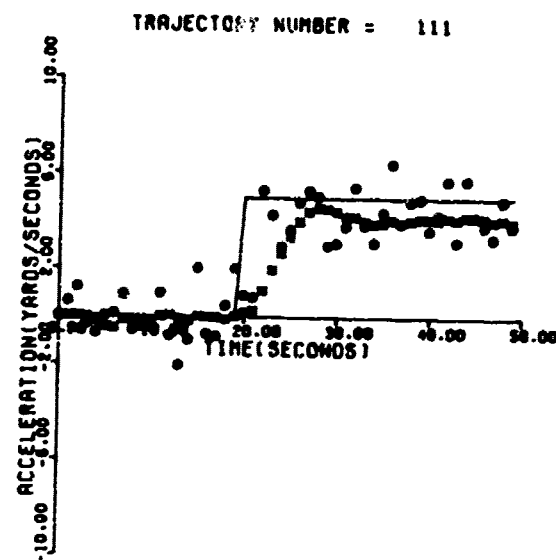
We will now consider a few particular examples of target motion—actually driven by target acceleration—that can lead either to a filter divergence problem or perhaps to an unacceptably wide-banded filter implementation. Using these examples, we will then demonstrate and discuss several possible techniques of filter adaptation to deal with the situation.

Let us first consider the random-acceleration filter model developed in Section IV. In that section we discussed the values of the parameters representing target maneuverability and maneuver frequency and found, as might be expected, that no single set of numbers would adequately represent the target scenario. Rather, it was found that a range of each parameter could be expected and that we can essentially bound the parameters by a low maneuver level, low-frequency (long time constant) parameter set which we will designate $(\sigma_{MA}, \omega_{MA})$ (where ω_M is $1/\tau_M$) and a high-maneuver-level, high-frequency set $(\sigma_{MB}, \omega_{MB})$. Our fundamental assumption in approaching adaptation with the random-acceleration model is therefore

$$(\sigma_{MA}, \omega_{MA}) \leq (\sigma_M, \omega_M) \leq (\sigma_{MB}, \omega_{MB}) \quad (5.1)$$

where (σ_M, ω_M) is the parameter set representing any actual target. Recall that a non-zero choice of $(\sigma_{MA}, \omega_{MA})$ is due to the fact that an actual target moving through an actual atmosphere will be buffeted by turbulence (and perhaps other effects) so that $(\sigma_{MA}, \omega_{MA})$ really might represent inadvertent maneuvers of the target. It is important that we do not allow the A parameters to vanish since steady-state error covariance would also vanish and the filter could diverge due even to the mildest maneuvers.

Let us now construct filters based upon each bounding parameter set and called the corresponding (fixed-parameter) Kalman filters A and B. We will now exercise these filters against several target profiles to assess their acceleration tracking performance. In Figure 5.1, the actual acceleration of several targets is shown by the solid lines. The asterisks plot the acceleration estimates of the A filter and the circles those of the B filter. The maneuver parameter sets used for these filters are $(\sigma_{MA}, \omega_{MA})$ of $(0.5, 1/20)$ and $(\sigma_{MB}, \omega_{MB})$ of $(5.0, 1/10)$. These acceleration profiles and their two integrals were generated on the computer, and numerically generated white noise with $\sigma = 5$ yards was added to the true positions to simulate the measurements. It is observed that the A filter is narrow-banded in that the acceleration estimates are relatively smooth. Unfortunately, the A filter values tend to diverge (or at least lag severely) from the true value whenever the acceleration changes rapidly. The B filter estimates tend to usually be unbiased (since filter B is very wide-banded) and never really divergent. Unfortunately, the B filter estimates always contain a large amount of noise, even when we see that acceleration can be tracked smoothly and accurately by the A filter. Figure 5.1 displays dramatically the classical problem of determining the proper filter bandwidth to yield the smoothest unbiased estimates. The paradox is that,



X Narrow (A)

O Wide (B)

Figure 5.1. Acceleration Tracking Performance: Examples for Narrow and Wide Bandwidth Filters

while each filter has its advantages and disadvantages, neither is really suitable (as they are) for implementation as a GFCS tracking filter. It is therefore clear that we should look for some method of adapting *on-line* the bandwidth of the filter to effect the desired performance.

Before leaving Figure 5.1, the reader might note that these particular acceleration profiles obviously belie our target model assumption that acceleration is a stationary, first-order random process. Indeed, many real targets will not follow this model any more than they might follow a polynomial model. The point again is that we want each trajectory to be reasonably represented by the statistics of the random process.

B. RESIDUAL STATISTICS AND MANEUVER DETECTION

When divergence occurs, the error vector (estimated state minus true state) grows large. Fortunately, we are able to monitor at least partially the actual performance of the filter at any given time. This is done by observing the sequence of residuals (often referred to as the "innovations sequence") and attempting to detect the buildup of a bias and consequent growth of the residuals. We can determine the statistics of the residuals by recalling the definition

$$\underline{v}(k/k-1) = \underline{z}(k) - H(k) \hat{x}(k/k-1)$$

and the measurement model

$$\underline{z}(k) = H(k) \underline{x}(k) + \underline{v}(k)$$

Substituting for $\underline{z}(k)$, we find

$$\underline{v}(k/k-1) = \underline{v}(k) - H(k) \underline{v}(k/k-1) \quad (5.2)$$

By taking expected values, we find immediately that

$$E [\underline{v}(k/k-1)] = 0 \quad (5.3)$$

and

$$E [\underline{v}(k/k-1) \underline{v}^T(k/k-1)] = R(k) + H(k) P(k/k-1) H^T(k) \quad (5.4)$$

Substituting the matrices for our model, we find that, when the filter is operating optimally, the residual sequence should be zero-mean Gaussian with variance

$$\sigma_v^2(k) = \sigma^2(k) + P_{11}(k/k-1) \quad (5.5)$$

It is then a relatively simple matter to determine the probability that the sampled residual belongs to the population with the above statistics.

An important extension of Equation (5.2) can be made in the case of the actual sampled residual. The actual residual is comprised of exactly the same error terms, and the *sample* expected value (denoted E_p) is related in the same manner to the *actual* error and measurement covariance. *i.e.*,

$$E_p [\underline{p}(k/k-1) \underline{p}^T(k/k-1)] = R_{ACT}(k) + H_{ACT}(k) P_{ACT}(k/k-1) H_{ACT}^T(k) \quad (5.6)$$

or for our case

$$\begin{aligned} E_p \{v^2(k/k-1)\} &= \sigma_{vACT}^2(k) \\ &= \sigma_{ACT}^2(k) + P_{11ACT}(k/k-1) \end{aligned} \quad (5.7)$$

If we are sufficiently confident in our estimate of the measurement error variance to assume that our filter estimate $\sigma^2(k)$ equals $\sigma_{ACT}^2(k)$, then we find that by using Equations (5.5) and (5.7) we might make certain important inferences about the validity of our calculated error covariance and, if necessary, adjust it accordingly. If our estimate of $\sigma_{ACT}^2(k)$ is very inaccurate, then adaptive techniques based upon the use of such information would be inadvisable. Another technique to avoid this problem will be discussed later but was not implemented.

Rather than work with an individual residual, greater statistical significance can be obtained by considering several data samples. We will use a sample mean which, defined for any variable f , is

$$\bar{f}(k) = \frac{1}{M} \sum_{i=k-M}^{i=k} f(i) \quad (5.8)$$

It is much more convenient to implement the sample mean recursively by means of a fixed memory length averager that will approximate the exact sample mean to $O(\Delta t)$. This recursive sample mean is given as

$$\bar{f}(k) = G_1 \bar{f}(k-1) + G_2 f(k) \quad (5.9)$$

where the constant gains are

$$G_1 = (M - 1)/M = 1 - G_2 \quad (5.10a)$$

$$G_2 = 1/M \quad (5.10b)$$

The effective memory length (or "window") of this averager is simply $T_M = M\Delta t$. Of course, this averager is not valid when $t < T_M$, but that does not really matter here. We will choose a value T_M that will minimize the maneuver detection time. The influence of T_M on maneuver detection time will be briefly considered shortly.

Using these equations as tools, it is now possible to construct various sample means involving the residuals and corresponding tests for each one. For example, we will normalize each residual with $\sigma_v(k)$ (to remove the transient nature of the error covariance) and compute the normalized sample mean as

$$\bar{v}_N(k) = \frac{1}{M} \sum_{i=k-M}^{i=k} v(i/i-1)/\sigma_v(i) \quad (5.11)$$

where the subscript N refers to the normalization. It is easily seen that \bar{v}_N ideally is also a zero-mean normally distributed random variable of variance $1/M$ that can provide an indication of actual filter performance relative to the calculated (assumed) performance. We could alternatively choose the normalized mean square residual defined as

$$\bar{v}_N^2(k) = \frac{1}{M} \sum_{i=k-M}^{i=k} v^2(i/i-1)/\sigma_v^2(i) \quad (5.12)$$

We find that $M \cdot \bar{v}_N^2$ is the chi-squared variable with M degrees of freedom. The expected value of \bar{v}_N^2 is unity and the variance is two. It is therefore a relatively easy problem to construct tests concerning these variables. Other maneuver detectors can be constructed by considering the correlation of residuals. For example, it can be shown that the autocovariance

$$\Omega_v(i) = E [v(k/k-1) v^T(k-i/k-1-i)]$$

should vanish for $i \neq 0$. This information forms the basis for a slightly different type of maneuver detection. For example, the adaptive filter in the MARK 86 GFCS monitors the signs of the residuals and declares a maneuver when a certain number of successive residuals show the same sign. Such methods work very well. There are other more complicated variations of this type. All of these maneuver detection statistics are similar in that each starts with the assumption that the residuals should be uncorrelated, zero-mean, Gaussian when the filter is operating properly. In fact, the author could find no particular advantage of

any one maneuver detector over another as far as performance is concerned. We have chosen, for reasons of computational efficiency, the normalized residual sample mean (Equation 5.11) to construct our criterion for maneuver detection. We can now define the maneuver detector.

We will define a maneuver as any *target motion* that causes the filter performance, as measured by \bar{v}_N , to exceed some specified value. Namely, a maneuver is declared if

$$|\bar{v}_N(k)| > C \sigma_{\bar{v}_N}(k) = C/\sqrt{M} \quad (5.13a)$$

or equivalently

$$\bar{v}_N^2(k) > C^2/M \quad (5.13b)$$

where C is a constant that determines the significance of the test. The probability of indicating a maneuver when there is none (a "false detection" or Type I error) is

$$P_{FD} = \{P \bar{v}_N^2(k) > C^2 \sigma_{\bar{v}_N}^2(k)\} \quad (5.14)$$

Values of P_{FD} as a function of C can be found in most introductory statistics books. For example, maneuver detection at the two- and three-sigma levels ($C = 2$ or 3) yields $P_{FD} = 4.56\%$ and 0.270% respectively. In order to choose a value of C , we must consider the false detection probability in conjunction with the cost—presumably in degraded performance—of such false detections. Such costs are a function of the type of action taken to adapt when a maneuver is declared. These costs will be considered, at least qualitatively, for the various adaptive techniques to be discussed. A tradeoff is involved since there is a cost in increased maneuver detection time when the maneuver threshold C is raised. It is possible to estimate analytically the functional dependence between maneuver detection time and the various parameters involved with detection for certain simple maneuvers. For example, let us consider the step acceleration target (of the type discussed previously) under the assumption that the filter is completely converged and perfectly nonresponsive, i.e., the filter error covariance and gains are zero. A step in acceleration a_x will cause a residual bias buildup of magnitude

$$v_{bias}(t_s) = \frac{1}{2} a_x t_s^2 \quad (5.15)$$

where t_s is the time since application of a_x . The bias of the normalized residual sample mean will then be

$$\bar{\nu}_{N \text{ bias}}(t_s) = \frac{1}{\sigma_v T_M} \int_0^{t_s} \frac{1}{2} a_s t^2 dt = \frac{a_s}{6 \sigma_v T_M} t_s^3 \quad (5.16)$$

Maneuver detection occurs at time T_D when

$$\bar{\nu}_{N \text{ bias}}(T_D) = C/\sqrt{M} \quad (5.17)$$

so that

$$T_D = (6 T_M C \sigma_v / \sqrt{M} a_s)^{1/3} \quad (5.18)$$

Using Equation (5.5) and recalling that P_{11} is assumed essentially zero, we can write

$$\sigma_v = \sqrt{\sigma^2 + P_{11}} \cong \sigma$$

Also so that

$$T_M / \sqrt{M} = \sqrt{M} \Delta t \quad (5.20)$$

$$T_D \cong (6 C \sigma \Delta t \sqrt{M} / a_s)^{1/3}$$

Similar procedures can be used to estimate maneuver detection time for other target models. For example, an acceleration ramp (jerque j_s) would yield a residual bias of

$$\nu_{\text{bias}}(t_s) = \frac{1}{6} j_s t_s^3 \quad (5.21)$$

and we find

$$T_D = (24 C \sigma \Delta t \sqrt{M} / j_s)^{1/4} \quad (5.22)$$

Equations such as (5.20) and (5.22) are admittedly not exact since the steady-state filter covariance and gains are not zero and the filter's response, however small, would tend to increase the time required for detection. This effect is small, however, for a narrow-bandwidth filter. The results, on the other hand, are quite interesting. The detection time for both cases is found to be directly proportional to (the fractional powers of) C , M and σ and inversely proportional to the magnitude of the step change. This is intuitively satisfying since we would expect it to take longer to detect a small maneuver than a large one or longer to detect a maneuver with a higher threshold, a longer memory length or when observed with the superposition of more noise. Since the probability of a false detection is a function only of C , it is seen that the residual average does not improve our maneuver

detector. Indeed, the residual mean serves only to increase the maneuver detection time without effect on the false detection rate. Therefore, the sample residual mean will be selected with $M = 1$, i.e., no memory length. If we had chosen another type of maneuver detector (such as estimating a residual trend), this would not necessarily be true. Notice also that, for the step acceleration target, an increase in the value of C , say from 1 to 3, increases the maneuver detection time less than 7 percent but improves the false detection rate by almost a factor of 17.

We have chosen Equation (5.13) as our maneuver detector. It can be shown to yield performance equivalent to any of the other methods discussed in this section. Results with this detector will be presented in the following sections.

C. VARIABLE BANDWIDTH ADAPTATION

Once divergence (or a "maneuver," as we call any target motion that produces divergence even temporarily) has been detected, a method of modifying or adapting the filter parameters to correct the situation must be specified. There are several methods of dealing with divergence, most of which effectively increase the gains to make the filter more sensitive to new data and, of course, more sensitive to noise. We will consider a brief survey of these various techniques (excluding parallel filtering to be discussed in the next section).

Several survey and comparison papers on adaptation appear in the literature. Mehra (1972) classified the different methods into four categories and discussed the relationships between them and the difficulties associated with each. Hagar (1973) conducted a fairly comprehensive investigation of the various types of adaptive algorithms and their capabilities in a very useful reference. Sidar and Bar-Shlomo (1972) simulated and compared a number of adaptive filters for application to their gyro compass problem. Particular emphasis was placed on the Jazwinski-type of covariance matching (to be discussed shortly).

There are several variants of a technique which the author refers to as bias correctors. Demetry and Titus (1968) suggested a bias corrector, whereby if a maneuver is detected one reprocesses the most recent data with a wider-bandwidth filter. Friedland (1969) devised a technique, whereby the state is augmented with a bias vector which is then estimated in an effectively decoupled estimator. McAulay and Denlinger (1973) suggested a multiple-order derivative polynomial filter whereby the lower order ($n = 1$) would be used unless a maneuver is detected. If a maneuver is detected, then a higher-order ($n = 2$) filter is initialized. Essentially this technique attempts to break the trajectory into piecewise polynomials with the break-points defined on line.

Other investigators have tried to directly adapt the memory length or the gain matrix itself. Epstein (1971) used the memory length of the filter as the adaptive variable and

varied the memory as a function of the residual series. Mehra (1972) presented a new algorithm for the direct estimation of the optimal gain. Hampton and Cooke (1973) also designed an adaptive filter for tracking high-performance maneuvering targets. This technique uses the orthogonality property of the residual sequence to automatically track the optimal gain levels.

There are some interesting algorithms that attempt to "learn" the dynamic as they process the data. Mehra (1971, #1) devised a technique to actually estimate *on line* essentially all of the (assumed) linear system, i.e., the order, the transition matrix, and the measurement and process-noise matrices, even when the processes are nonstationary (of a certain type). The author has not actually tried this algorithm but strongly suspects there would not be enough time or computational capacity to use this technique. Of course, one could never be certain that, once a strategy based on past data was determined, the strategy would continue to be employed in the future. But, of course, that factor is merely a part of the difficulty faced by the predictor designer in any case.

Divergence can be considered as having been caused by an inaccurate estimate of the noise covariances. It is only natural then that attempts are made to estimate these covariances as well as the state. Mehra (1970) introduced a method to simultaneously estimate Q and R when the state model was assumed. Weiss (1970) surveyed and discussed techniques of this same type. The principal objective of these algorithms is to effect a correspondence between the actual covariance and the calculated covariance--hence the name for these methods, "covariance matching." Nahi (1972) and Soeda and Yoshimura (1973) developed procedures to more or less optimally modify the calculated error covariance to prevent divergence when the residual is not likely to have come from the calculated distribution. A pioneer of the covariance matching technique, Jazwinski (1969), developed the concept (usually known by his name) of preventing divergence by covering modeling errors with noise and adaptively estimating the noise level. The Jazwinski method, at least in a conceptual level, is the approach selected for our application. It is particularly well suited for our purpose since the random acceleration model tells us much about the proper structure of the noise process and its relation to the target kinematics.

The Jazwinski technique uses "residual feedback" to specify the proper level of process noise to add. The process works as follows. In the event of a maneuver detection, we make use of Equation (5.6) (assuming accurate knowledge of the measurement noise covariance) to estimate the level of actual process noise. One assumes that the actual extrapolated error covariance consists of three terms, i.e.,

$$P_{ACT}(k/k-1) = \phi(k_1 k-1) P(k-1/k-1) \phi^T(k k-1) + Q(k-1) + Q^*(k-1) \quad (5.23)$$

where the first two terms on the righthand side represent the calculated error covariance with a small process noise covariance Q assumed by the filter. The term Q^* is effectively

added to balance the equation. If there is no maneuver detected, Q^* is zero. It should be noted that it is not possible to uniquely specify Q^* with this equation. We can only determine HQ^*H^T or Q_{11}^* for our case. This is not particularly troublesome as we can choose some normalized form of the Q^* matrix such as the random acceleration model process noise matrix to automatically define the other elements of Q^* as a function of Q_{11}^* . For our case, we could calculate

$$Q_{11}^*(k-1) = [\bar{v}_N^2(k) - 1/M] [\sigma^2(k) + P_{11}(k/k-1)] \quad (5.24)$$

when a maneuver was detected, calculate the remaining elements of Q^* with the Singer form, and add Q^* to the error covariance. This technique was simulated and was found to yield fairly good adaptive filter performance. Unfortunately, this adaptive filter would occasionally display erratic behavior and generate unreasonable error covariances. In order to eliminate this problem and properly constrain the error covariance, a different means of specifying Q^* was selected.

We therefore return to our original assumption that the target maneuver level--and subsequent process noise--is bounded by the A and B parameter sets. Therefore, in the absence of maneuver detection, process noise corresponding to set A will be added and, if a maneuver is declared, Q_B , corresponding to set B, will be added.

That is:

$$\text{if } [\bar{v}_N^2(k) < C^2/M], \text{ then } Q(k-1) = Q_A, \quad (5.25a)$$

$$\text{if } [\bar{v}_N^2(k) > C^2/M], \text{ then } Q(k-1) = Q_B. \quad (5.25b)$$

If a maneuver is declared, some time will elapse before Q_B builds up the error covariance matrix. Another obvious alternative is to simply reset the error covariance in order to obtain a faster response. This technique was rejected, however, since one must pay a very large price for a false detection. Using Equation (5.25), it is entirely possible for the maneuver detector to turn off before the error covariance builds up to the steady-state value. Consequently, Figure 5.2 shows the sequence of events. Initially, the target is not maneuvering and the filter, using the A parameters, is tracking acceleration very well. After the maneuver occurs, an interval of time elapses before the maneuver is detected which we have been referring to as $T_{\text{detection}}$. Once the maneuver is detected and we start adding Q_B , it takes another amount of time, called T_{adapt} , for the filter to "open" or to increase the error covariance sufficiently to remove the bias. When the maneuver detector decides the bias is gone and turns off, the filter returns to adding the small process noise Q_A . The time required to return to steady state with the A parameters is called $T_{\text{reconverge}}$.

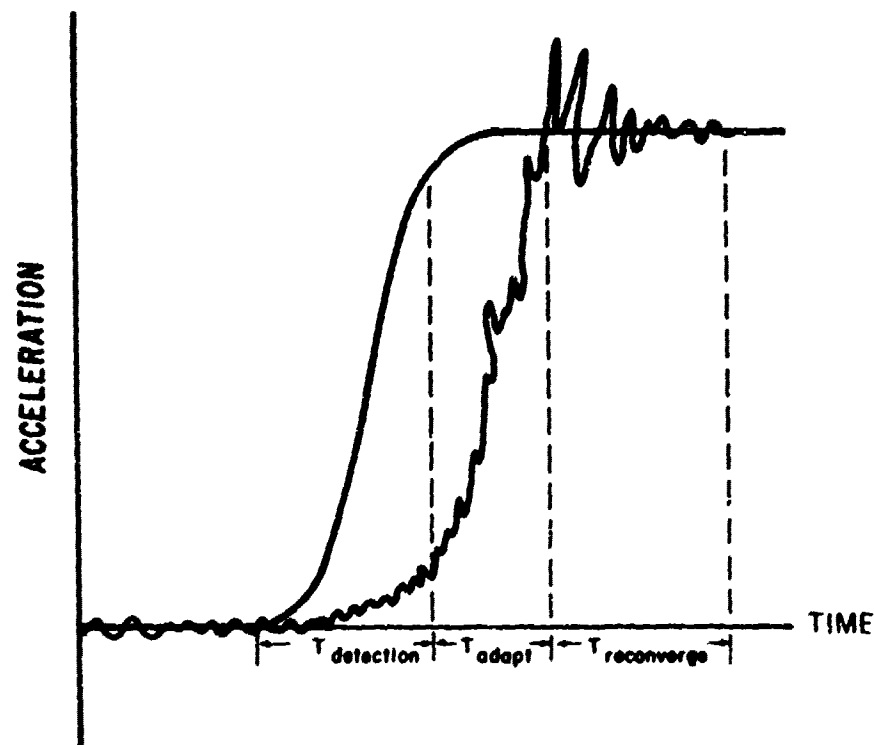


Figure 5.2. Conceptual Example: Single Variable-Band Adaption

In Figure 5.3, the example acceleration profiles are shown being tracked by the single, adaptive, variable-bandwidth filter. The maneuver parameter sets are the same as those used in the previous nonadaptive examples. The maneuver detector was operated at the $C = 3$ (sigma) level to minimize false detections and no memory ($M = 1$) to minimize detection time. It is observed that this technique tracks rather smoothly while improving the bias error. There tends to be some overshoot when the filter adapts because it was initially lagging and builds up excessive rates in order to catch up. The filter appears to respond well (as expected) to the step change in acceleration. The predicted detection time of 1.65 sec. apparently matches the simulation quite well. The variable-bandwidth or adaptive filter represents a marked improvement in accuracy over either of the fixed-bandwidth filters discussed previously.

D. DUAL BANDWIDTH ADAPTATION

Upon considering the single-variable-bandwidth filter just discussed, it was felt that certain improvements could be made in the adaptive performance. Ideally, when a maneuver has been detected, one would likely reprocess the measurements over some interval

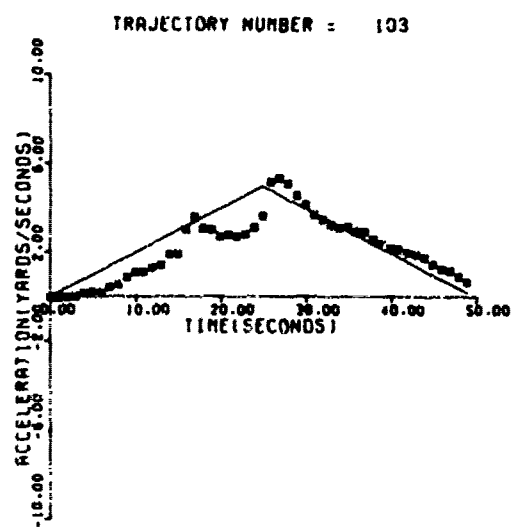
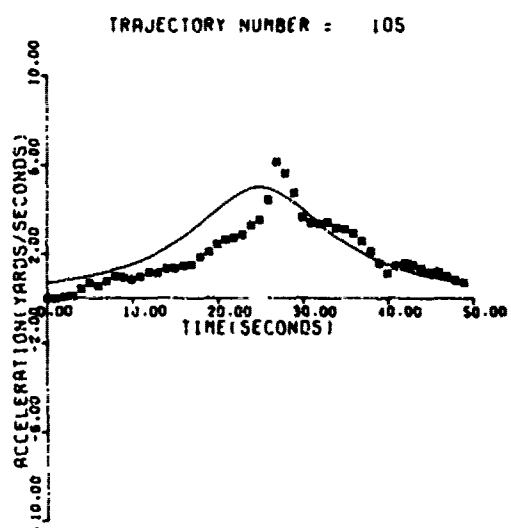
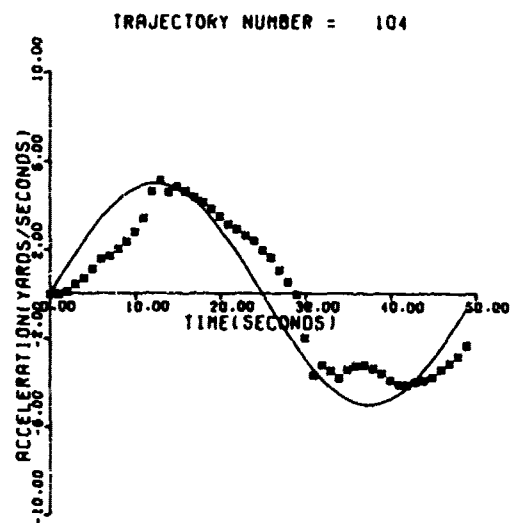
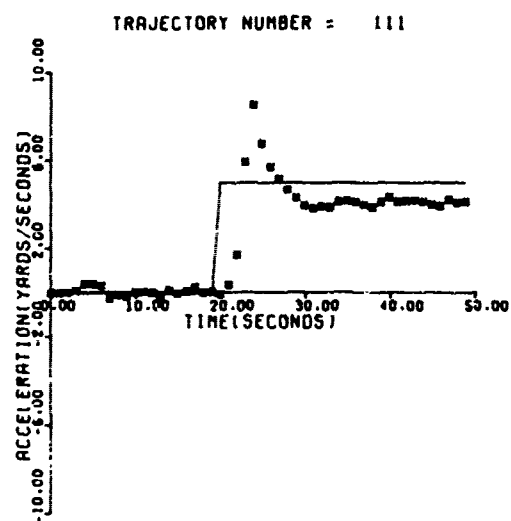


Figure 5.3. Acceleration Tracking Performance: Examples with Adaptive Variable Bandwidth Filter

immediately preceding current time with a wider-bandwidth filter so as to remove the bias error which presumably has been occurring. Unfortunately, our application essentially demands a fully recursive filter in order to efficiently implement the same on a limited real-time computer. It would be very difficult to interrupt normal processing in order to re-process past data. The best answer obviously is some type of recursive wide-bandwidth filter operating in parallel to the "main filter" which can be used to (more or less) instantaneously remove the bias error once it has been detected. Such a technique has the obvious advantage of eliminating the last two waiting intervals of the single variable bandwidth filter—the adaptation and reconvergence times. If done properly, it can also eliminate the overshoot after adaptation.

The dual-bandwidth filter would work as follows. Two filters, A and B, corresponding to the respective maneuver parameter bounds, would operate simultaneously. The filter would only output (to the FCS) the state vector of filter A, \hat{x}_A . If divergence of filter A is detected, using the detection criterion of Equation (5.25), the state vector of filter B, \hat{x}_B , which should be unbiased, is put into filter A, i.e.,

$$\text{If } [\bar{p}_{\hat{x}_A}^2(k) > C^2/M], \text{ then } \hat{x}_A = \hat{x}_B \quad (5.26)$$

Conceptually, we want the adaptation to work as in Figure 5.4. The A filter is outputting smooth estimates of the state until the maneuver is detected. Using Equation (5.26), the A filter then "jumps" to the current (unbiased) estimate of the B filter and no adaption or reconvergence is required. Admittedly, the output vector \hat{x}_A will be discontinuous, but in this situation \hat{x}_B represents the best information available. An important consideration was the decision as to what modification, if any, should be made to the A filter bandwidth. Three options were considered and tested.

Theoretically, if one resets the state estimate of A to that of B, the bandwidth should be similarly reset. That is, if a maneuver is declared and Equation (5.26) is in effect, then

$$\text{Option 1: } P_A = P_B$$

This option was found unacceptable, however, as one pays a high cost of a false detection since the long reconvergence time has not been eliminated. Leaving the A bandwidth unchanged, i.e.,

$$\text{Option 2: } P_A = P_A$$

was desirable since it did not suffer the disadvantages of Option 1. The particular maneuver detector we are using, however, did not do well with Option 2 since the large random errors of the B filter look like biases to the A filter and repeated maneuver detections tended to occur. In order to eliminate this problem, the author decided on

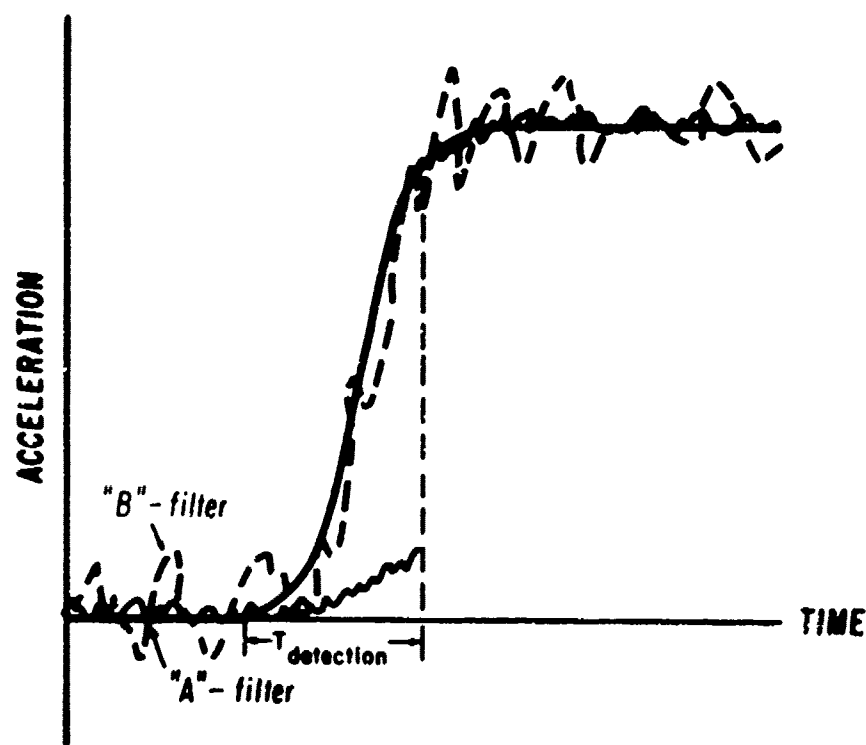


Figure 5.4. Conceptual Example: Dual-Bandwidth Adaptation

$$\text{Option 3: } P_A = P_A + Q_B \quad (5.27)$$

i.e., a gradual widening of the A filter bandwidth which eliminated the cyclic detections and maintained a low cost of a false detection. In Figure 5.5, the same acceleration profiles were simulated with the same parameters for the maneuver statistics and maneuver detector but with the dual-bandwidth filter. The acceleration estimation errors were reduced even further and the overshoot errors disappeared. The root-mean-square acceleration estimation error was reduced on the average by 25 percent over the single-bandwidth filter.

Another interesting possibility, only superficially examined by the author to date, involves the generation of an output vector, \hat{x}_0 , which is always a linear combination of \hat{x}_A and \hat{x}_B : i.e.,

$$\hat{x}_0 = W \hat{x}_A + (1 - W) \hat{x}_B \quad (5.28)$$

where the weighting factor W is a function of the residual statistics--both sample and calculated--of both the A and B filter. An obvious advantage of such a technique would be to construct a (more-or-less) optimal combination of the A and B filter with continuity of

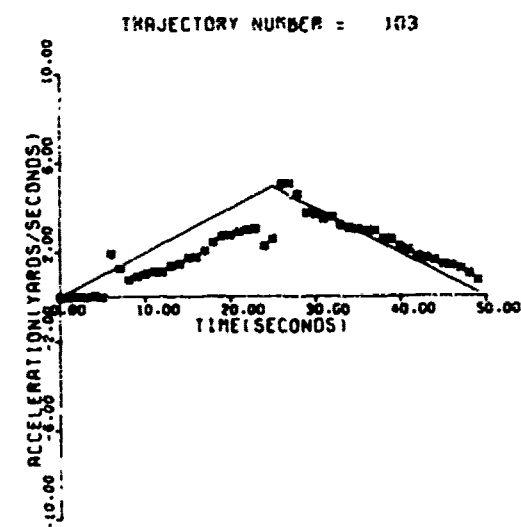
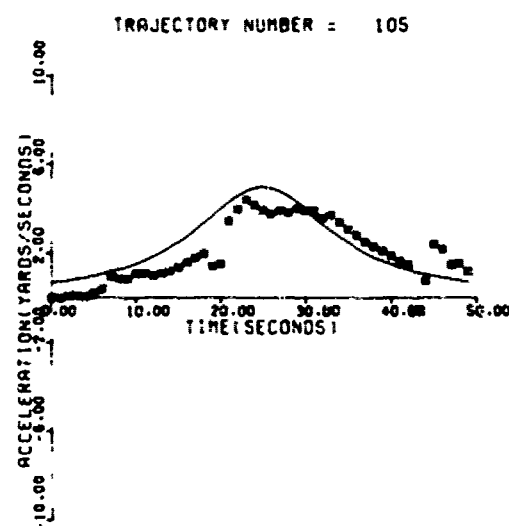
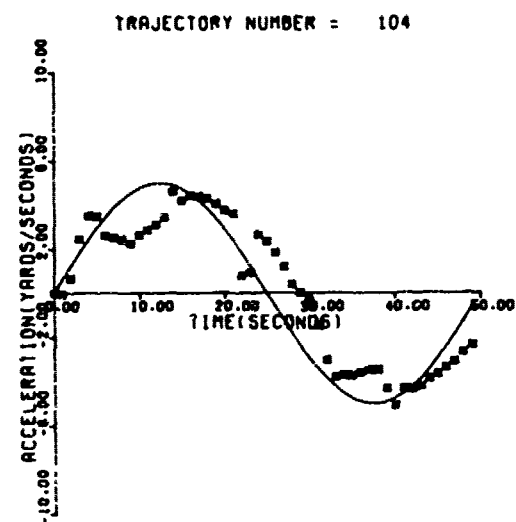
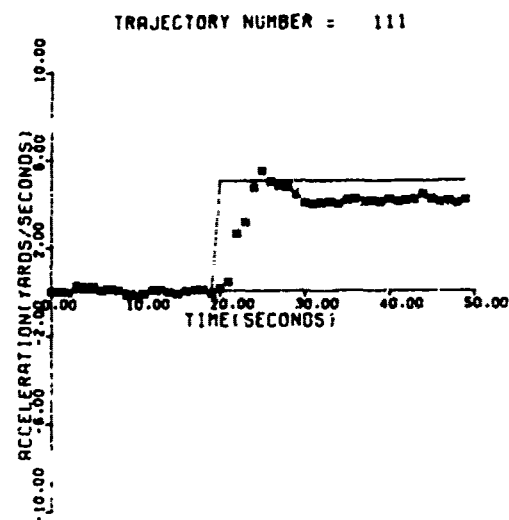


Figure 5.5. Acceleration Tracking Performance: Examples with Adaptive Dual Bandwidth Filter

the output state vector simply a by-product not without merit of its own. Thorpe (1973) developed a dual-bandwidth filter concept in which target maneuvers, assumed to be relatively infrequent events were modeled by introducing a binary random variable in the target state equation. He used the likelihood ratio for the detection of a maneuver to determine the weight W . Brown and Price (1974) studied the utility of a bank of parallel filters, each with a different bandwidth, using the random acceleration model. A combined estimate was constructed on the basis of a hypothesis test of the probability that each filter is the correct one. Alspach (1973) also constructs a bank of parallel filters and uses Bayesian techniques to estimate the optimal gain. Moose (1972) also uses Bayesian methods to determine the relative weighting of the output of a bank of Kalman filters to form the best combined estimate. The author believes these techniques deserve further consideration for implementation in a GFCS application.

In Equation (5.7), we related the sample residual variance to the actual measurement and estimation error variance and mentioned the problem that if our estimate of the measurement variance is poor then maneuver detection based upon that equation would not work well. To illustrate, let us consider the sensitivity of the adaptation process to *large* errors in our estimate of measurement error. If we greatly underestimate the measurement error level, then the sample mean residual will often exceed the expected value due to measurement error, and false maneuver detections will result. Unfortunately, such false detections lead to a filter bandwidth change in just the opposite direction of the proper adaptation. Instead of increasing the gains and thereby weighting the measurements heavier, we should increase R , thereby decreasing the gains. Conversely, if we grossly overestimate the measurement error variance, then maneuver detection is delayed (perhaps indefinitely) so that we do not increase the gains to follow the target. For the application intended for this filter, this problem should not occur, as good estimates of the sensor statistics should be available in order to properly model them.

Another approach which the author wishes to pursue is to statistically analyze directly the residual sequence for each filter of a parallel filter bank without regard to the calculated and/or assumed statistics. In other words, we simply look at each residual sequence and pick the smoothest unbiased one. This technique could also be incorporated with the weighting technique of Equation (5.28). The author has experimented with such a residual-analysis/weighting-factor approach but has not yet determined a method which works well. Brown and Price (1974) also tried a variant of this approach with reasonable success. The principal advantage of such a technique, if workable, would be that it has essentially no sensitivity to an incorrect estimate of the measurement statistics.

In summary, the author has tried or at least studied many different adaptation techniques but has to date found none that works any better than the one presented. There appear, however, to be a large number of alternatives in the literature (some of them mentioned in this section) that offer promise of improvement and which should be investigated

further. The concept of a bank of filters of various bandwidths is particularly appealing. Sensitivity to incorrect observation error statistics is probably the most serious problem that must be addressed. This problem will be discussed in a future report.

VI. SERIALY CORRELATED MEASUREMENT ERROR

In Section II, the conventional discrete Kalman filter was described. An important and necessary assumption for that development was that the measurement error be "white" or uncorrelated. In Section III, we found that it is desirable to process data at the highest rate possible when the measurement errors are independent. Unfortunately, the assumption of white noise and the desirability of a high data rate are often incompatible when using measurements from real physical systems. For example, truly white noise never actually exists in an real system. When sampling data from such a system, eventually one usually finds the noise autocorrelated (serially correlated) when observed at some sufficiently high data rate.

In this section, we will address this problem. We will present a common noise model and a modification to the conventional Kalman filter that deals with this model. We will find that one can quantitatively assess the degradation in performance due to autocorrelation and can perform simple sensitivity analyses to determine the effects of incorrect estimates of the statistical parameters.

A. NOISE MODEL

The measurement model for the Kalman filter, as presented in Section II, is

$$\underline{z}(k) = H(k) \underline{x}(k) + \underline{v}(k)$$

where it was assumed that

$$E [\underline{v}(k)] = 0$$

and

$$E [\underline{v}(j) \underline{v}^T(k)] = R(k) \delta_{jk}$$

We will now replace the assumption of the last equation with the more general assumption that the measurement noise \underline{v} is the output of a linear discrete dynamic system driven by white noise. I.e.,

$$\underline{v}(k) = \Psi(k, k-1) \underline{v}(k-1) + \underline{\xi}(k-1) \quad (6.1)$$

where

$$E [\underline{\xi}(k)] = 0 \quad (6.2)$$

and

$$E [\underline{\xi}(j) \underline{\xi}^T(k)] = R^*(k) \delta_{jk} \quad (6.3)$$

Sage and Melsa (1971) point out, "Although this is not the most general form of colored noise, it is probably the most general practical form. One often has extreme difficulty in establishing the parameters of such a simple model, so that it is hard to consider the use of any more general form." It is easy to determine that

$$R(k) = \Psi(k, k-1) R(k-1) \Psi^T(k, k-1) + R^*(k-1) \quad (6.4)$$

and that the autocovariance of v is

$$E [\underline{v}(k) \underline{v}^T(k-1)] = \Psi(k, k-1) R(k-1) \quad (6.5)$$

These equations are useful in defining Ψ and R^* .

Now let us consider a particular type of first-order process chosen for the measurement error model for this application. An exponential autocorrelation function is found to be both convenient and reasonably matches power spectral density information that has been estimated for candidate sensors for the MARK 68 GFCS. The noise propagation equation for this case is:

$$v(k) = \rho(k) v(k-1) + \sigma_w(k) \xi(k-1) \quad (6.6)$$

where

$$\rho(k) = \exp [-\Delta t / \tau(k)] \quad (6.7)$$

is the correlation coefficient for lag t . The standard deviation of the white noise driving term is

$$\sigma_w(k) = \sigma(k) \sqrt{1 - \rho^2(k)} \quad (6.8)$$

and

$$\sigma(k) = E [v^2(k)] \quad (6.9)$$

as before. We will be primarily concerned with constant values of τ between 0 and 0.30 seconds. Although we will consider parametrically a wider range of values, note that this process is not, strictly speaking, assumed to be stationary or Gaussian although, in actuality, it is only slowly varying and slightly non-Gaussian.

A digital noise generation program, required for simulation purposes, is discussed and presented in Appendix A. The algorithm, based upon a paper by L. F. Balas (1967), will generate noise of the type assumed by Equations (6.6) to (6.8).

B. RESTRUCTURED KALMAN FILTER

In this section, the details of applying an algorithm to process data with autocorrelated measurement noise (of the type discussed in the previous paragraphs) will be presented. The algorithm, from Sage and Melsa (1971), is reproduced in Table 6.1. Before substituting the matrices for our system into the algorithm, it was found convenient to rearrange the given equations for our purposes. The original form of the algorithm consists of two processes: smoothing and filtering. The author found that it is possible to rewrite these equations so that the algorithm appears to be of an extrapolation-update form which, of course, is the method the conventional Kalman filter in Section II is written in. It was found that this form requires less computation than the original and can be easily related to the white noise filter. It is pertinent to note at this time that the Sage and Melsa algorithm was chosen over the augmented state approach since the former does not require an increase in the dimension of the state vector and does not result in ill-conditioned computations. The authors principally responsible for the algorithm in Sage and Melsa are Bryson and Henrikson (1968) who discuss the relative merits of the two approaches.

Let us now consider the smoother/filter algorithm in Table 6.1. Notice first that the residual for the smoothing and filtering equations, Equations (8) and (9), are identical. We will define this new *a priori* residual as

$$\underline{v}^*(k/k-1) = \underline{z}(k) - \Psi(k, k-1) \underline{z}(k-1) - H^*(k-1) \hat{\underline{x}}(k-1/k-1) \quad (6.10)$$

If we substitute the definition of $H^*(k-1)$ from Equation (7), we find we can write

$$\underline{v}^*(k/k-1) = \underline{v}(k/k-1) - \Psi(k, k-1) \underline{v}(k-1/k-1) \quad (6.11)$$

where

$$\underline{v}(k/k-1) = \underline{z}(k) - H(k) \underline{x}(k/k-1) \quad (6.12)$$

is the usual residual for the white noise filter and

Table 6.1. Smoother/Filter Algorithm for Autocorrelated Measurement Noise

MODEL

$$\underline{x}(k+1) = \phi(k+1, k) \underline{x}(k) + \underline{w}(k) \quad (1)$$

OBSERVATIONS

$$\underline{z}(k) = H(k) \underline{x}(k) + \underline{v}(k) \quad (2)$$

$$\underline{v}(k+1) = \Psi(k+1, k) \underline{v}(k) + \underline{\xi}(k) \quad (3)$$

STATISTICS

$$E[\underline{w}(k)] = E[\underline{\xi}(k)] = E[\underline{w}(j) \underline{\xi}^T(k)] = 0 \quad (4)$$

$$E[\underline{w}(j) \underline{w}^T(k)] = Q(k) \delta_{jk} \quad (5)$$

$$E[\underline{\xi}(j) \underline{\xi}^T(k)] = R^*(k) \delta_{jk} \quad (6)$$

DEFINITION

$$H^*(k-1) = H(k) \phi(k, k-1) - \Psi(k, k-1) H(k-1) \quad (7)$$

SMOOTHER

$$\begin{aligned} \hat{\underline{x}}(k-1/k) = & \hat{\underline{x}}(k-1/k-1) + K_s(k-1) [\underline{z}(k) \\ & - \Psi(k, k-1) \underline{z}(k-1) - H^*(k-1) \hat{\underline{x}}(k-1/k-1)] \end{aligned} \quad (8)$$

FILTER

$$\begin{aligned} \hat{\underline{x}}(k/k) = & \phi(k, k-1) \hat{\underline{x}}(k-1/k) + K_f(k) [\underline{z}(k) \\ & - \Psi(k, k-1) \underline{z}(k-1) - H^*(k-1) \hat{\underline{x}}(k-1/k-1)] \end{aligned} \quad (9)$$

Table 6.1. Smoother/Filter Algorithm for Autocorrelated Measurement Noise--(Continued)

SMOOTHER GAIN

$$K_s(k-1) = P(k-1/k-1)H^{*T}(k-1)[H^*(k-1)P(k-1/k-1)H^{*T}(k-1) + R^*(k-1) + H(k)Q(k-1)H^T(k)]^{-1} \quad (10)$$

FILTER GAIN

$$K_f(k) = Q(k-1)H^T(k)[H^*(k-1)P(k-1/k-1)H^{*T}(k-1) + R^*(k-1) + H(k)Q(k-1)H^T(k)]^{-1} \quad (11)$$

SMOOTHED COVARIANCE

$$P(k-1/k) = [I - K_s(k-1)H^*(k-1)]P(k-1/k-1) \\ [I - K_s(k-1)H^*(k-1)] + K_s(k-1)[R^*(k-1) + H(k)Q(k-1)H^T(k)]K_s^T(k-1) \quad (12)$$

FILTERED COVARIANCE

$$P(k/k) = \phi(k, k-1)P(k-1/k)\phi^T(k, k-1) + Q(k-1) \\ - K_f(k)[H^*(k-1)P(k-1/k-1)H^{*T}(k-1) + R^*(k-1) \\ + H(k)Q(k-1)H^T(k)]K_f^T(k) - \phi(k, k-1)K_s(k-1)H(k)Q(k-1) \\ - Q(k-1)H^T(k)K_s^T(k-1)\phi^T(k, k-1) \quad (13)$$

$$\underline{g}(k-1/k-1) = \underline{z}(k-1) - H(k-1)\underline{\hat{x}}(k-1/k-1) \quad (6.13)$$

is the *a posteriori* residual from the previous filter cycle. We will now define

$$\eta(k) = \Psi(k, k-1)\underline{g}(k-1/k-1) \quad (6.14)$$

which is essentially the correlated portion of the old residual that we want to remove. Therefore, Equation (6.11) becomes

$$\underline{p}^*(k/k-1) = \underline{p}(k/k-1) - \eta(k) \quad (6.15)$$

Since the smoother and filter equations are of similar form, it is easily seen that the two equations can be combined to produce "extrapolation" and "update" equations, identical in form to the white noise filter. I.e.,

$$\underline{\hat{x}}(k/k-1) = \phi(k, k-1) \underline{\hat{x}}(k-1/k-1) \quad (6.16)$$

and

$$\underline{\hat{x}}(k/k) = \underline{\hat{x}}(k/k-1) + \underline{K}^*(k) \underline{p}^*(k/k-1) \quad (6.17)$$

where

$$\underline{K}^*(k) = \phi(k, k-1) \underline{K}_s(k-1) + \underline{K}_f(k) \quad (6.18)$$

is the equivalent gain.

Now let us consider the gain matrices. Notice that the inverse of the matrix

$$\underline{G}(k) = \underline{H}^*(k-1) \underline{P}(k-1/k-1) \underline{H}^{*T}(k-1) + \underline{R}^*(k-1) + \underline{H}(k) \underline{Q}(k-1) \underline{H}^T(k) \quad (6.19)$$

appears in both gain equations. I.e.,

$$\underline{K}_s(k-1) = \underline{P}(k-1/k-1) \underline{H}^{*T}(k-1) \underline{G}^{-1}(k) \quad (6.20)$$

and

$$\underline{K}_p(k) = \underline{Q}(k-1) \underline{H}^T(k) \underline{G}^{-1}(k) \quad (6.21)$$

Therefore

$$\underline{K}^*(k) = [\phi(k, k-1) \underline{P}(k-1/k-1) \underline{H}^{*T}(k-1) + \underline{Q}(k-1) \underline{H}^T(k)] \underline{G}^{-1}(k) \quad (6.22)$$

By defining the extrapolated covariance as before, i.e.,

$$\underline{P}(k/k-1) = \phi(k, k-1) \underline{P}(k-1/k-1) \phi^T(k, k-1) + \underline{Q}(k-1) \quad (6.23)$$

we can then write

$$\underline{K}^*(k) = [\underline{P}(k/k-1) \underline{H}^T(k) - \underline{A}(k)] \underline{G}^{-1}(k) \quad (6.24)$$

and

$$G(k) = H(k) P(k/k-1) H^T(k) + R^*(k-1) + B(k) \quad (6.25)$$

where

$$A(k) = \phi(k, k-1) P(k-1/k-1) H^T(k-1) \Psi^T(k, k-1) \quad (6.26)$$

and

$$\begin{aligned} B(k) &= \Psi(k, k-1) H(k-1) P(k-1/k-1) H^T(k-1) \Psi^T(k, k-1) \\ &- H(k) \phi(k, k-1) P(k-1/k-1) H^T(k-1) \Psi^T(k, k-1) \\ &- \Psi(k, k-1) H(k-1) P(k-1/k-1) \phi^T(k, k-1) H^T(k) \end{aligned} \quad (6.27)$$

Obviously, if Ψ vanishes, then both A and B vanish and $R^* = R$. We therefore recover the original form of the white noise gain equation.

Now we will consider the error covariance equations. The smoothed error covariance equation, Equation (12), can be rewritten in a much simpler form by expanding the first term and recombining, making use of the gain equation, i.e.,

$$\begin{aligned} P(k-1/k) &= P(k-1/k-1) + K_s(k-1) H^*(k-1) P(k-1/k-1) H^{*T}(k-1) K_s^T(k-1) \\ &- P(k-1/k-1) H^{*T}(k-1) K_s^T(k-1) + K_s(k-1) H^*(k-1) P(k-1/k-1) \\ &+ K_s(k-1) [R^*(k-1) + H(k) Q(k-1) H^T(k)] K_s^T(k-1) \\ &= P(k-1/k-1) + K_s(k-1) G(k) K_s^T(k-1) - \psi(k-1/k-1) H^{*T}(k-1) K_s^T(k-1) \\ &- K_s(k-1) H^*(k-1) P(k-1/k-1) \\ &= [1 - K_s(k-1) H^*(k-1)] P(k-1/k-1) \end{aligned} \quad (6.28)$$

Using the smoothing gain equation and the fact that P and G are symmetric, we can also write this equation as

$$P(k-1/k) = P(k-1/k-1) - K_s(k-1) G(k) K_s(k-1) \quad (6.29)$$

Using the filter gain equation, we can also simplify the filtered covariance equation.

$$\begin{aligned}
P(k/k) &= \phi(k, k-1) P(k-1/k) \phi^T(k, k-1) + Q(k-1) - Q(k-1) H^T(k) K_f^T(k) \\
&\quad - \phi(k, k-1) K_s(k-1) H(k) Q(k-1) H^T(k) Q(k-1) H^T(k) K_s^T(k-1) \phi^T(k, k-1) \\
&= \phi(k, k-1) P(k-1/k) \phi^T(k, k-1) + Q(k-1) - Q(k-1) H^T(k) K^{*T}(k) \\
&\quad - \phi(k, k-1) K_s(k-1) H(k) Q(k-1)
\end{aligned} \tag{6.30}$$

By substituting Equation (6.29) for the smoothed covariance and then for the smoothing gain, we find

$$\begin{aligned}
P(k/k) &= \phi(k, k-1) P(k-1/k-1) \phi^T(k, k-1) + Q(k-1) \\
&\quad - [\phi(k, k-1) K_s(k-1)] G(k) [\phi(k, k-1) K_s(k-1)]^T \\
&\quad - Q(k-1) H^T(k) K^{*T}(k) - \phi(k, k-1) K_s(k-1) H(k) Q(k-1) \\
&= P(k/k-1) - [K^*(k) - Q(k-1) H^T(k) G^{-1}(k)] G(k) [K^*(k) - Q(k-1) H^T(k) G^{-1}(k)]^T \\
&\quad - Q(k-1) H^T(k) K^{*T}(k) - [K^*(k) - Q(k-1) H^T(k) G^{-1}(k)] H(k) Q(k-1) \\
&= P(k/k-1) - K^*(k) G(k) K^{*T}(k)
\end{aligned} \tag{6.31}$$

which is of a form identical to the white noise filter. A summary of the equations for the equivalent extrapolation/update algorithm is found in Table 6.2. Upon comparing this algorithm with the white noise filter (Table 2.1), we find that the correlated noise algorithm requires the additional calculation of $A(k)$, $B(k)$ and $\eta(k)$. A FORTRAN IV version of this algorithm, called CORKAL, was written by the author and can be found in Appendix A. We now apply this algorithm to our model.

Our model will now consist of the current version of the dynamics and prediction, as developed in the previous sections, but we will add the correlated noise model. The measurement noise transition matrix is

$$\Psi(k, k-1) = \Psi = [\rho(k)] \tag{6.32}$$

The standard measurement noise covariance matrix R will be replaced by the white measurement noise.

$$R^*(k) = [\sigma_w^2(k)] \tag{6.33}$$

Upon substituting these matrices into the algorithm, we find that the complicated matrix expressions reduce to rather simple algebraic calculations for our problem. The matrix A is (3×1) , and B and η are scalars. The algorithm with the indicated modifications can be found again summarized in Table 6.3. (None of the filter algorithms considered in this section are adaptive. The adaptive versions are exactly analogous to the white noise models in Section V.)

Table 6.2. Equivalent Extrapolation/Update Algorithm
for Autocorrelated Measurement Noise

MODEL

$$\underline{x}(k+1) = \phi(k+1, k)\underline{x}(k) + \underline{w}(k) \quad (1)$$

OBSERVATIONS

$$\underline{z}(k) = H(k)\underline{x}(k) + \underline{v}(k) \quad (2)$$

$$\underline{v}(k+1) = \Psi(k+1, k)\underline{v}(k) + \underline{\xi}(k) \quad (3)$$

STATISTICS

$$E[\underline{w}(k)] = E[\underline{\xi}(k)] = E[\underline{w}(j)\underline{\xi}^T(k)] = 0 \quad (4)$$

$$E[\underline{w}(j)\underline{w}^T(k)] = Q(k)\delta_{jk} \quad (5)$$

$$E[\underline{\xi}(j)\underline{\xi}^T(k)] = R^*(k)\delta_{jk} \quad (6)$$

STATE EXTRAPOLATION

$$\hat{\underline{x}}(k/k-1) = \phi(k, k-1)\hat{\underline{x}}(k-1/k-1) \quad (7)$$

COVARIANCE EXTRAPOLATION

$$A(k) = \phi(k, k-1)P(k-1/k-1)H^T(k-1)\Psi^T(k, k-1) \quad (8)$$

Table 6.2. Equivalent Extrapolation/Update Algorithm
for Autocorrelated Measurement Noise--(Continued)

$$\begin{aligned} B(k) &= \Psi(k, k-1)H(k-1)P(k-1/k-1)H^T(k-1)\Psi^T(k, k-1) \\ &\quad - H(k)\phi(k, k-1)P(k-1/k-1)H^T(k-1)\Psi^T(k, k-1) \\ &\quad - \Psi(k, k-1)H(k-1)P(k-1/k-1)\phi^T(k, k-1)H^T(k) \end{aligned} \quad (9)$$

$$P(k/k-1) = \phi(k, k-1)P(k-1/k-1)\phi^T(k-1) + Q(k-1) \quad (10)$$

GAIN

$$G(k) = H(k)P(k/k-1)H^T(k) + R^*(k-1) + B(k) \quad (11)$$

$$K^*(k) = \{P(k/k-1)H^T(k) - A(k)\}G^{-1}(k) \quad (12)$$

STATE UPDATE

$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + K^*(k)[\underline{z}(k) - H(k)\hat{\underline{x}}(k/k-1) - \underline{\eta}(k)] \quad (13)$$

COVARIANCE UPDATE

$$P(k/k) = P(k/k-1) - K^*(k)G(k)K^{*T}(k) \quad (14)$$

A POSTERIORI RESIDUAL

$$\eta(k) = \Psi(k, k-1)[\underline{z}(k) - H(k-1)\hat{\underline{x}}(k-1|k-1)] \quad (15)$$

Table 6.3. Filter Equations for Correlated Measurement Noise: A Summary

STATE EXTRAPOLATION

$$\hat{x}_1(k/k-1) = \hat{x}_1(k-1/k-1) + \Delta t \hat{x}_2(k-1/k-1) + \alpha \hat{x}_3(k-1/k-1) \quad (1)$$

$$\hat{x}_2(k/k-1) = \hat{x}_2(k-1/k-1) + \beta \hat{x}_3(k-1/k-1) \quad (2)$$

$$\hat{x}_3(k/k-1) = \gamma \hat{x}_3(k-1/k-1) \quad (3)$$

CORRELATION MATRICES

$$A_1(k) = \rho(k)[P_{11}(k-1/k-1) + \Delta t P_{12}(k-1/k-1) + \alpha P_{13}(k-1/k-1)] \quad (4)$$

$$A_2(k) = \rho(k)[P_{12}(k-1/k-1) + \beta P_{13}(k-1/k-1)] \quad (5)$$

$$A_3(k) = \rho(k)\gamma P_{13}(k-1/k-1) \quad (6)$$

$$B(k) = \rho^2(k)P_{11}(k-1/k-1) - 2A_1(k) \quad (7)$$

COVARIANCE EXTRAPOLATION

$$\begin{aligned} P_{11}(k/k-1) = & P_{11}(k-1/k-1) + 2\Delta t P_{12}(k-1/k-1) + 2\alpha P_{13}(k-1/k-1) \\ & + \Delta t^2 P_{22}(k-1/k-1) + 2\alpha \Delta t P_{23}(k-1/k-1) + \alpha^2 P_{33}(k-1/k-1) \\ & + Q_{11}(k-1) \end{aligned} \quad (8)$$

$$\begin{aligned} P_{12}(k/k-1) = & P_{12}(k-1/k-1) + \beta P_{13}(k-1/k-1) + \Delta t P_{22}(k-1/k-1) \\ & + (\alpha + \beta \Delta t) P_{23}(k-1/k-1) + \alpha \beta P_{33}(k-1/k-1) + Q_{12}(k-1) \end{aligned} \quad (9)$$

$$\begin{aligned} P_{13}(k/k-1) = & \gamma P_{13}(k-1/k-1) + \gamma \Delta t P_{23}(k-1/k-1) + \alpha \gamma P_{33}(k-1/k-1) \\ & + Q_{13}(k-1) \end{aligned} \quad (10)$$

$$\begin{aligned} P_{22}(k/k-1) = & P_{22}(k-1/k-1) + 2\beta P_{23}(k-1/k-1) + \beta^2 P_{33}(k-1/k-1) \\ & + Q_{22}(k-1) \end{aligned} \quad (11)$$

Table 6.3. Filter Equations for Correlated Measurement Noise: A Summary—(Continued)

$$P_{23}(k/k-1) = \gamma P_{23}(k-1/k-1) + \beta \gamma P_{33}(k-1/k-1) + Q_{23}(k-1) \quad (12)$$

$$P_{33}(k/k-1) = \gamma^2 P_{33}(k-1/k-1) + Q_{33}(k-1) \quad (13)$$

GAINS

$$G(k) = P_{11}(k/k-1) + \sigma_w^2(k-1) + B(k) \quad (14)$$

$$K_1^*(k) = [P_{11}(k/k-1) - A_1(k)]/G(k) \quad (15)$$

$$K_2^*(k) = [P_{12}(k/k-1) - A_2(k)]/G(k) \quad (16)$$

$$K_3^*(k) = [P_{13}(k/k-1) - A_3(k)]/G(k) \quad (17)$$

RESIDUAL

$$\nu^*(k/k-1) = z(k) - \hat{x}_1(k/k-1) - \eta(k) \quad (18)$$

STATE UPDATE

$$\hat{x}_1(k/k) = \hat{x}_1(k/k-1) + K_1^*(k)\nu^*(k/k-1) \quad (19)$$

$$\hat{x}_2(k/k) = \hat{x}_2(k/k-1) + K_2^*(k)\nu^*(k/k-1) \quad (20)$$

$$\hat{x}_3(k/k) = \hat{x}_3(k/k-1) + K_3^*(k)\nu^*(k/k-1) \quad (21)$$

$$\eta(k+1) = \rho(k+1) [z(k) - \hat{x}_1(k/k)] \quad (22)$$

COVARIANCE UPDATE

$$P_{11}(k/k) = P_{11}(k/k-1) - G(k)K_1^*(k)K_1^*(k) \quad (23)$$

$$P_{12}(k/k) = P_{12}(k/k-1) - G(k)K_1^*(k)K_2^*(k) \quad (24)$$

$$P_{13}(k/k) = P_{13}(k/k-1) - G(k)K_1^*(k)K_3^*(k) \quad (25)$$

$$P_{22}(k/k) = P_{22}(k/k-1) - G(k)K_2^*(k)K_2^*(k) \quad (26)$$

Table 6.3. Filter Equations for Correlated Measurement Noise: A Summary--(Continued)

$$P_{23}(k/k) = P_{23}(k/k-1) - G(k)K_2^*(k)K_3^*(k) \quad (27)$$

$$P_{33}(k/k) = P_{33}(k/k-1) - G(k)K_3^*/k)K_3^*(k) \quad (28)$$

INITIALIZATION

$$\hat{x}_1(0/0) = z(0) \quad (29)$$

$$\hat{x}_2(0/0) = \hat{x}_3(0/0) = 0 \quad (30)$$

$$P_{11}(0/0) = \sigma^2(0) \quad (31)$$

$$P_{12}(0/0) = 0. \quad (32)$$

$$P_{13}(0/0) = 0. \quad (33)$$

$$P_{22}(0/0) = \sigma_2^2(0) \quad (34)$$

$$P_{23}(0/0) = 0. \quad (35)$$

$$P_{33}(0/0) = \sigma_3^2(0) \quad (36)$$

CONSTANTS

$$\gamma = \exp(-\Delta t/\tau_M) \quad (37)$$

$$\beta = \tau_M(1 - \gamma) \quad (38)$$

$$\alpha = \tau_M^2(\gamma + \Delta t/\tau_M - 1) \quad (39)$$

$$\rho = \exp(-\Delta t/\tau) \quad (40)$$

PROCESS NOISE (CONSTANT)

$$Q_{33} = 2\Delta t\sigma_M^2/\tau_M \quad (41)$$

Table 6.3. Filter Equations for Correlated Measurement Noise: A Summary--(Continued)

$$Q_{23} = \Delta t Q_{33}/2 \quad (42)$$

$$Q_{22} = 2\Delta t Q_{23}/3 \quad (43)$$

$$Q_{13} = Q_{22}/2 \quad (44)$$

$$Q_{12} = 3\Delta t Q_{13}/4 \quad (45)$$

$$Q_{11} = 2\Delta t Q_{12}/5 \quad (46)$$

PREDICTION

$$\begin{aligned} \hat{x}_1(t + t_p/t) = & \hat{x}_1(t/t) + \hat{x}_2(t/t)t_p + \hat{x}_3(t/t)\tau_p^2 [\exp(-t_p/\tau_p) \\ & + t_p/\tau_p - 1] \end{aligned} \quad (47)$$

C. RESULTS AND SENSITIVITY

We will now apply this algorithm to assess the effect of measurement correlation on filter performance. The latest version of the filter and predictor, as developed in previous sections, remains the same in form. The correlated noise filter simply adds some additional terms. It should be noticed that the case of $\tau = 0$ (white noise) recovers the identical results as before. Figure 6.1 presents the results for this case and also those with several other values of up to $\tau = 1.0$ second. As expected, the presence of autocorrelation in the measurement errors tend to degrade filter performance both from the standpoint of settling time and steady state values. This is explained by the fact that the data being processed is not as informative (due to the presence of the random bias) as independent data. In fact, the performance degradation is rather severe when the correlation time is greater than 0.1 second. Of course, the amount of serial correlation is a function of the particular sensor system and, therefore, not a parameter under our direct control. From a software standpoint, we simply have to live with whatever sensor autocorrelation and subsequent performance degradation with which we are presented. It is hoped, however, that this consideration will be properly weighed, undoubtedly along with many others, when decisions as to a choice of sensor suits are effected.

For the one-dimensional work, values of the measurement error standard deviation of $\sigma = 5$ and 20 yards were assumed and used throughout the report in all sections that deal with only one-dimensional filtering. In fact, we have always normalized our criteria $\sigma_1(t + 10/t)$ by σ as if $\sigma_1(t + 10/t)$ were then independent of σ . While this is true for certain cases (see Section III, for example), it is not exactly true in general. Therefore, in

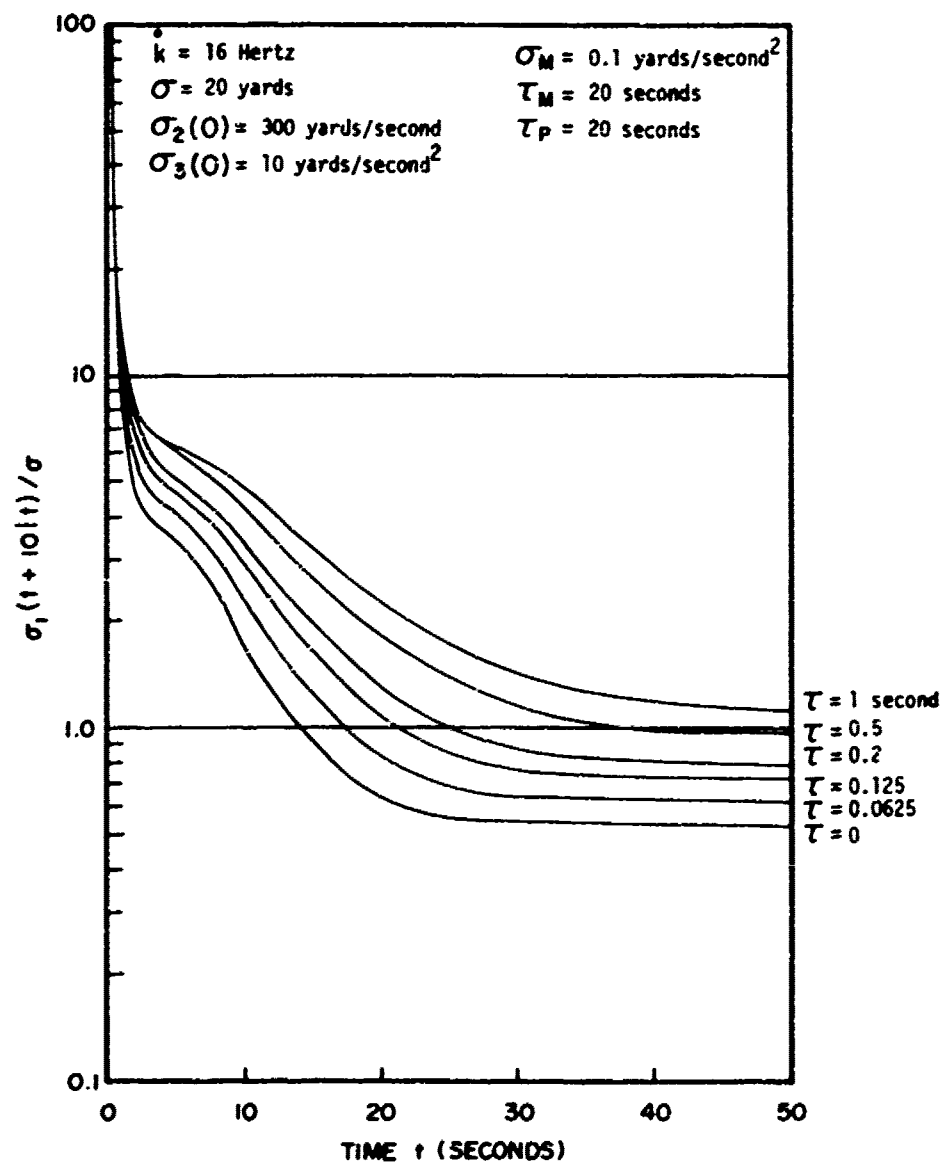


Figure 6.1. Correlated Measurement Noise Filter with Various Noise Correlation Times τ Normalized Standard Deviation of Predicted (10 second) Position Error

this section where we are considering the effect of one uncontrollable sensor parameter τ , it seems appropriate to consider another, i.e., σ . In Figure 6.2, we have therefore plotted a few cases with different parameters in order to see the effect. Examples with white noise ($\tau = 0$) and one with colored noise ($\tau = 0.30$ seconds) were chosen. As can be seen, the effect is not large but is not negligible. We find that the larger the value of σ , the smaller the steady-state value of $\sigma_1(t + 10/t)$. This indicates that the actual performance criteria, $\sigma_1(t + 10/t)$, has less than a one-to-one sensitivity to σ .

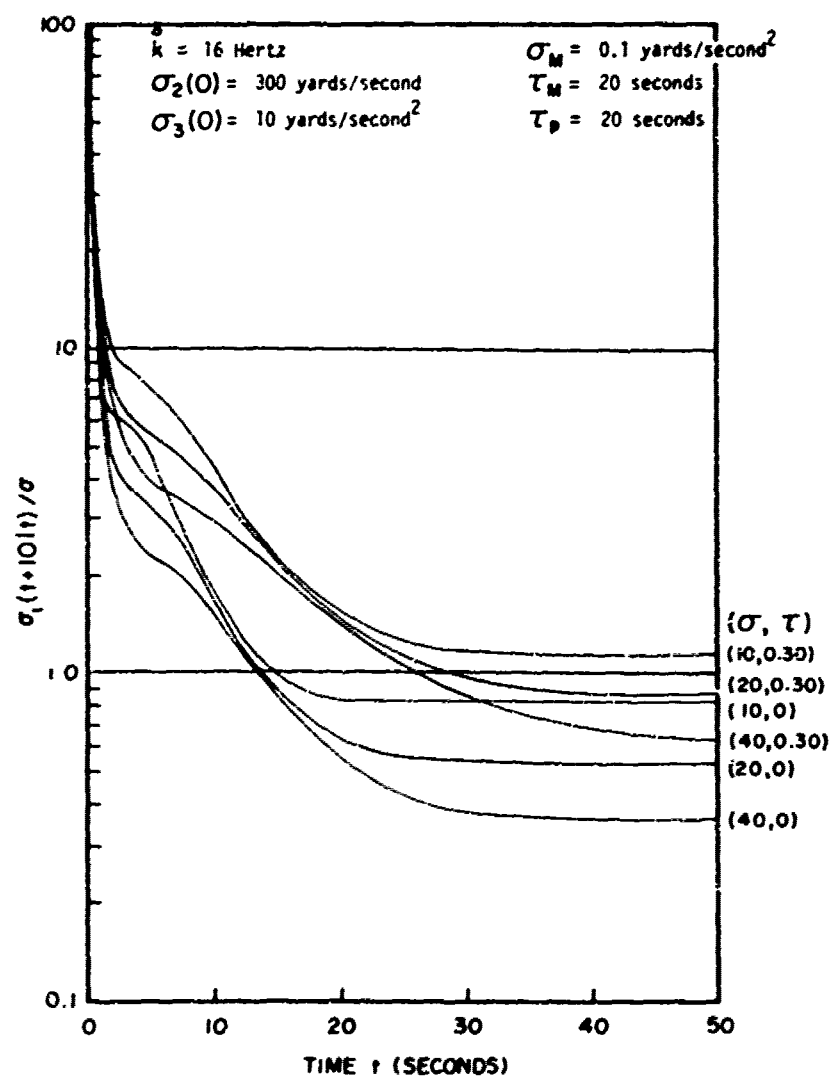


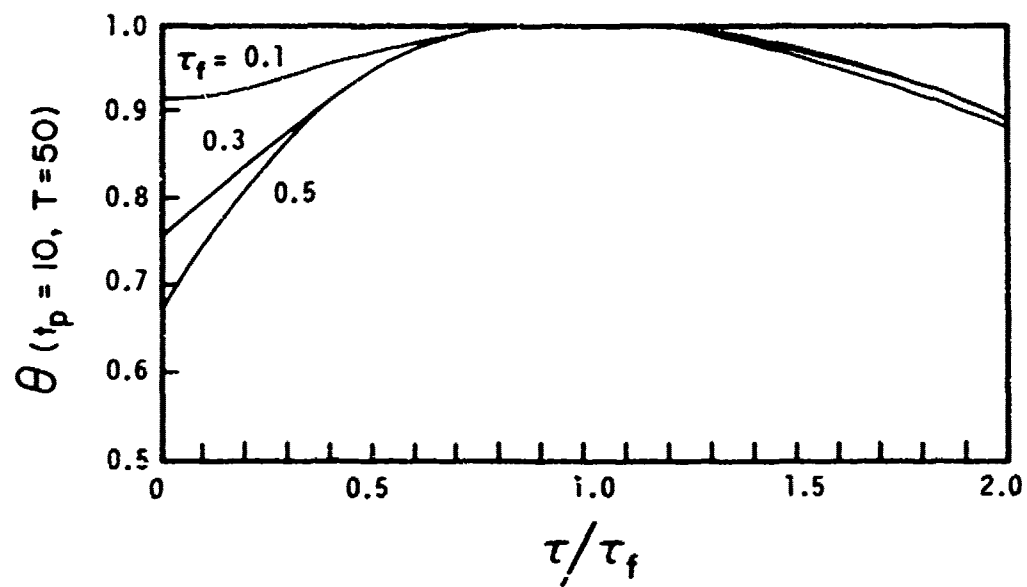
Figure 6.2. Correlated Measurement Noise Filter and Various Combinations of σ and τ Normalized Standard Deviation of Predicted (10 second) Position Error

While previously discussing the question of a reasonable range of values for τ , we alluded to the fact that we cannot always expect τ to be the same constant. In fact, we should expect our estimates of the sensor statistics to merely approximate the average behavior of the sensor system. When the sensor system is operated under various conditions and tuned by various personnel, we should expect some variability from our estimates to be realized. We then want to consider the degradation in filter performance--i.e., the sensitivity when the actual sensor parameters are other than those assumed by the filter. In order to do this, we must again calculate the value of the actual covariance when the gains are computed suboptimally (with the wrong values of τ and σ denoted by the subscript f) and compare this covariance with the optimal covariance. For the serially correlated measurement error filter, the actual covariance is propagated by Equation (2.16) in the extrapolation stage (the same as the white noise filter) and by the following equation for the update.

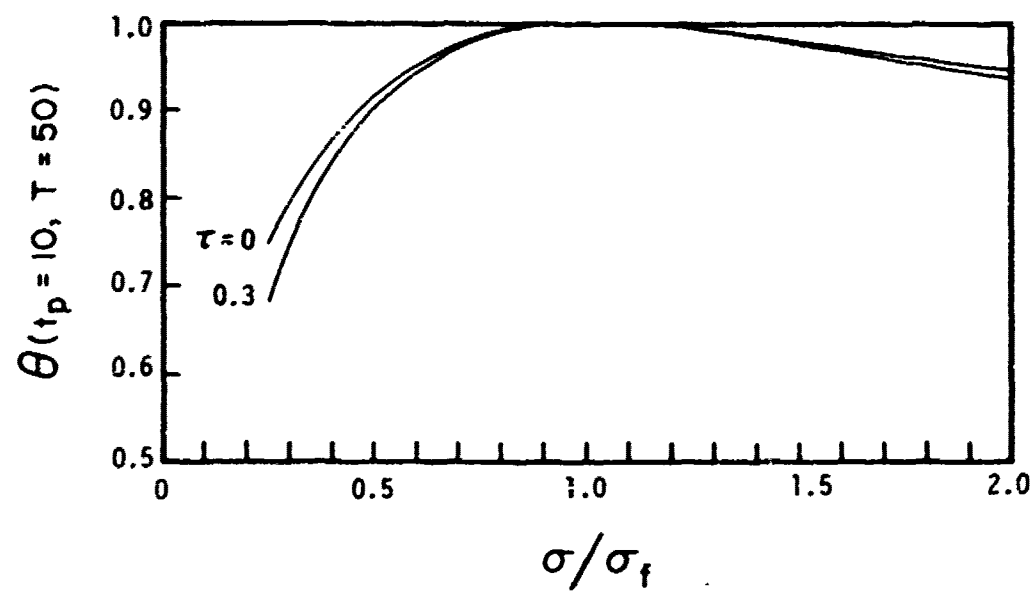
$$\begin{aligned} P_{ACT}(k/k) = & P_{ACT}(k/k-1) - [P_{ACT}(k/k-1) H_{ACT}^T(k) - A_{ACT}(k)] K^T(k) \\ & - K(k) [H_{ACT}(k) P_{ACT}(k/k-1) - A_{ACT}^T(k)] \\ & + K(k) G_{ACT}(k) K^T(k) \end{aligned} \quad (6.34)$$

where $A_{ACT}(k)$, $B_{ACT}(k)$ and $G_{ACT}(k)$ are defined in Table 6.2 with the actual values. We again use θ , as defined in Equation (4.17), as our figure of merit to measure optimality.

In Figure 6.3, θ is plotted for our standard problem $t_p = 10$ seconds integrated for $T = 50$ seconds. In (a), the ratio of the actual to assumed value of τ is allowed to vary from 0 to 2 for several assumed values of τ_f . Similarly in (b), the ratio of the actual assumed value of σ is allowed to vary from 0.25 to 2.00 for both white noise ($\tau = 0$) and colored noise ($\tau = 0.30$). In both cases, we find greater sensitivity when we overestimate the parameters ($\tau_f > \tau$ and $\sigma_f > \sigma$) and only very moderate performance degradation when we underestimate. In either case, it takes a severe error in the sensor parameter estimates to seriously degrade the Kalman filter. This is somewhat surprising and comforting since such filters tend to rely rather heavily on *a priori* estimates. We conclude that it is desirable not to overestimate τ or σ .



a. Sensitivity to an Incorrect Correlation Time Constant



b. Sensitivity to an Incorrect Measurement Standard Deviation

Figure 6.3.

Before leaving the subject of autocorrelation noise, let us consider one more question. It is interesting to determine the cost involved if autocorrelated noise were processed by a white noise filter ($\tau_f = 0$). With this information in hand, we also want to consider the relative performance of the effective white noise, suggested by D'Appolito (1971). D'Appolito says that an equivalent white noise of variance

$$\sigma_{\text{eff}}^2 = \left(\frac{1 + \rho}{1 - \rho} \right) \sigma^2 \quad (6.35)$$

"will produce the same estimation error as the original first order Markov process of variance σ^2 . Notice the factor multiplying σ^2 is always greater than one." In Figure 6.4, θ is again plotted for the case of $\tau_f = 0$ (white noise filter) when the actual noise has correlation time τ . We find that, if the measurement error standard derivation assumed by the white noise filter σ_f is equal to the nominal value σ , the performance drops off rather sharply when τ is greater than Δt (1/16 second here). Unfortunately, this analysis also indicates that the effective white noise ($\sigma_f = \sigma_{\text{eff}}$) performs no better and, in fact, considerably worse as the

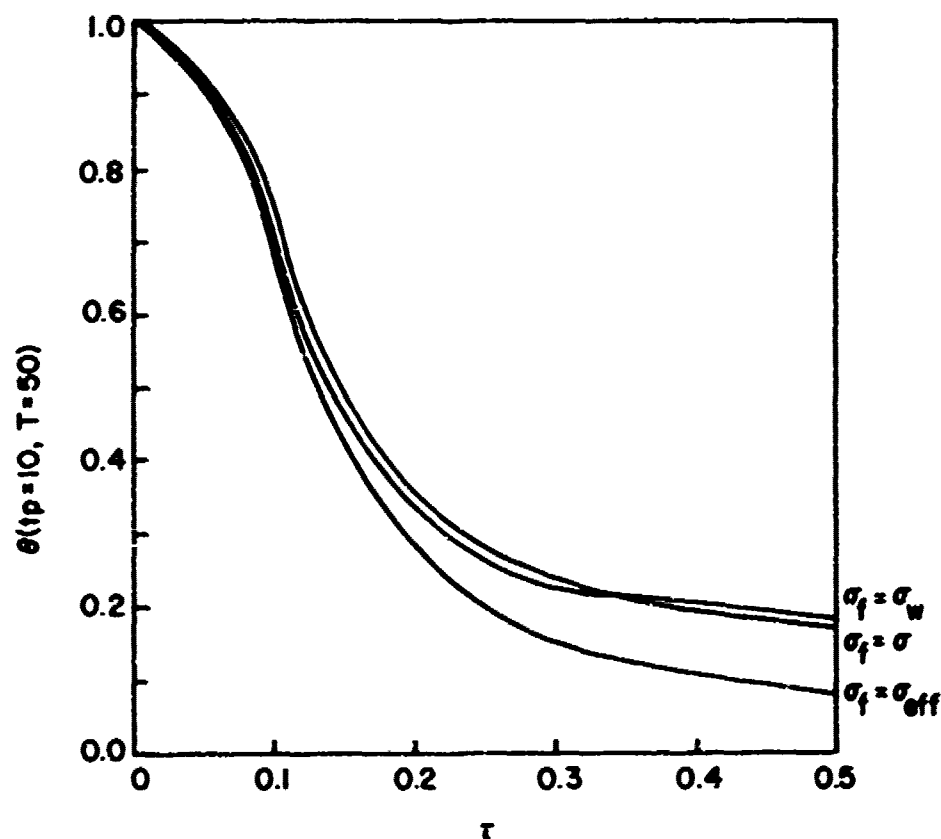


Figure 6.4. Performance of White Noise Filter with Various Values of σ

correlation time grows. Since increasing the nominal measurement error does not help, as an experiment the variance of the white noise driving term for the first-order Markov process was tried. We see from Equation (6.9), that σ_w is less than the nominal value σ . This value ($\sigma_f = \sigma_w$) produces about the same results as for $\sigma_f = \sigma$. We conclude, therefore, that near-optimal performance ($\theta = 0.9$), for the white noise filter is obtained only when $\tau < \Delta t$. For $\tau > \Delta t$, the white noise filter performance drops rapidly. It does not appear from this analysis that a white noise filter with an equivalent white noise (other than $\sigma_f = \sigma$) will help.

A final decision on the use of this correlated measurement noise algorithm must depend upon the final selection of a sensor suite chosen for the fire control system.

VII. TECHNIQUES FOR THE REDUCTION OF COMPUTATIONAL BURDEN

Several aspects of the filter developed to this point, such as a desire for a high data rate and real-time covariance propagation, begin to impose a significant computational burden on a small digital computer that might be utilized in a fire control system. Further development of the three-dimensional filter (in Section IX) and the addition of the dual-bandwidth adaptive features (in Section V) serve to multiply these computational requirements. Obviously, anything that can be done to reduce the calculations required by the basic one-dimensional filter could be significant. In this section, two possibilities are explored.

The high cycling rate of the filter might possibly be offset by data compression or "prefiltering," i.e., processing data at a higher rate than the filter cycles if this can be done without significantly degrading performance. The real-time propagation of error covariance on a fixed-point computer of limited word length (say 16 bits) poses other problems. Considering the potential range of the elements of the error covariance matrix, which involves the squares of both rather large and rather small numbers during one run, it was found that such a computer would have to perform the covariance calculations in double precision in order to allow sufficient scaling. There is also the possibility of the loss of the positive definite property, required for the error covariance, due to numerical error that results from finite word length calculations. Such a condition is disastrous as it usually leads to instability and ultimately total failure of the filter. The possibility of this occurring for our case is rather remote, however, since process noise is always added which tends to place a well-defined lower bound on the steady-state error covariance. These problems can be eliminated through the introduction of an error covariance square root which can be propagated in place of the covariance. There are also other possibilities for determining P , such as the information matrix (the inverse of error covariance) or its square root, but these will not be explored in this report.

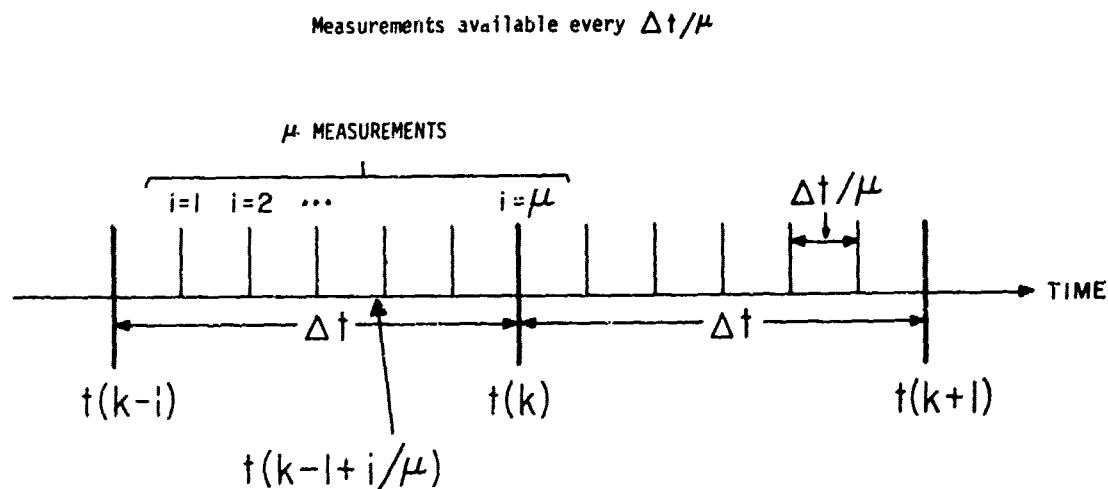
Another very important possibility, currently under investigation and not included in this report, is the possibility of using functional approximations for the error covariance

matrix. Such functions would be computed in the laboratory prior to implementation and would be functions of time as well as all the parameters (i.e., Δt , σ , σ_M , and τ_M). The approach presently being studied would work as follows. The error covariance matrix is calculated (via the Kalman filter equations) as a function of time and over a range of all the parameter space and stored on the computer. The steady-state solution is first fit using least squares over the parameter range. (Work is also under way to solve exactly the steady-state nonlinear matrix Riccati equation for P but a solution has not yet been obtained.) Then, knowing the steady-state solution and the initial conditions, the transient phase is fit as a function of time. The resulting functions are then used to calculate the gain matrix in the usual fashion. A sensitivity analysis is performed in order to assure an accurate approximation for the optimal gain. If this work is a success, most of the computational burden of the real-time Kalman filter—which is due to the error covariance equations—can be eliminated. The author is quite optimistic as to the future of this method and its application to the GFCS filtering problem.

A. PREFILTERS

Prefiltering, as the term will be defined in this report, means the processing of data which is available at a rate higher than that at which we wish to cycle the Kalman filter. It is also commonly referred to as data compression. Suppose, as shown in Figure 7.1, that the Kalman filter is cycled once every Δt seconds but that we wish to process data at an integer rate μ times the filter cycling rate. We will therefore have μ measurements, equally spaced $\Delta t/\mu$ apart, that will have been made since the last filter cycle at time $t(k-1)$ and which we want to process at time $t(k)$. There are undoubtedly several possible methods of aggregating (or lumping) or otherwise smoothing these additional measurements. However, we will consider (for now at least) the simplest effective method of doing this, namely, averaging.

As Warren (1974) points out, for short time intervals where the measurement noise essentially masks any time variation in the signal, data averaging is an effective means of data compression with small loss of information. Actually, we are not quite (but almost) in the region of applicability of this finding. For example, if we want to compress measurements from 16 to 4 Hertz, there is actually some velocity information that could be extracted from the measurements. The variance of this velocity estimate, however, is so large, relative to that already available in the Kalman filter, that its inclusion makes essentially no improvement. There is, however, a significant increase in the computation required to process such a velocity "measurement." We will therefore compute an equivalent prefiltered measurement based upon a technique similar to data averaging—residual averaging. By averaging the *a priori* residual as opposed to the measurements we account for the (estimated) target motion over the prefilter interval.



Kalman filter cycles every Δt

Figure 7.1. Prefilters--Relation of Measurement and Filtering Timing

The averaged (or prefiltered) residual is simply

$$\nu_{pr}(k/k-1) = \frac{1}{\mu} \sum_{i=1}^{i=\mu} \nu(k-1 + i/\mu | k-1) \quad (7.1)$$

All notation in this section on prefilters will be referenced to the filter cycling rate so that times will be in fractions (i/μ) of the time between filter cycles, Δt . Substituting the definition for the *a priori* residual, we find

$$\begin{aligned} \nu_{pr}(k, k-1) &= \frac{1}{\mu} \sum_{i=1}^{i=\mu} [z(k-1 + i/\mu) - \hat{x}_1(k-1 + i/\mu | k-1)] \\ &= \bar{z}(k) - \frac{1}{\mu} \sum_{i=1}^{i=\mu} \hat{x}_1(k-1 + i/\mu | k-1) \end{aligned} \quad (7.2)$$

where

$$\bar{z}(k) = \frac{1}{\mu} \sum_{i=1}^{i=\mu} z(k-1 + i/\mu) \quad (7.3)$$

is the average measurement. It is convenient to express the estimates \hat{x}_1 at times $(i/\mu) \Delta t$ relative to the extrapolated value at time t_k . This is done using the transition matrix and can easily be found to be

$$\begin{aligned} \hat{x}_1(k-1 + i/\mu | k-1) &= \hat{x}_1(k/k-1) + \left(\frac{i}{\mu} - 1\right) \Delta t \hat{x}_2(k/k-1) \\ &+ \alpha_i \hat{x}_3(k/k-1) \end{aligned} \quad (7.4)$$

where

$$\alpha_i = \alpha \left[\left(\frac{i}{\mu} - 1 \right) \Delta t \right] = \tau_M^2 \left[\gamma^{\left(\frac{i}{\mu} - 1 \right)} + \left(\frac{i}{\mu} - 1 \right) \Delta t / \tau_M - 1 \right] \quad (7.5)$$

Substituting Equation (7.4) into (7.2) and taking the indicated sums, we find

$$\begin{aligned} v_{pr}(k/k-1) &= \bar{z}(k) - \hat{x}_1(k/k-1) + \left[1 - \frac{1}{\mu^2} \sum_{i=1}^{i=\mu} i \right] \Delta t \hat{x}_2(k/k-1) \\ &- \left(\frac{1}{\mu} \sum_{i=1}^{i=\mu} \alpha_i \right) \hat{x}_3(k/k-1) \end{aligned} \quad (7.6)$$

We therefore define the *effective prefiltered measurement* as

$$z_{pr}(k) = \bar{z}(k) + \mu_2 \hat{x}_2(k/k-1) - \mu_3 \hat{x}_3(k/k-1) \quad (7.7)$$

where

$$\mu_2 = \Delta t \left(1 - \frac{1}{\mu^2} \sum_{i=1}^{i=\mu} i \right) \quad (7.8)$$

and

$$\mu_3 = \tau_M^2 \left[\frac{1}{\mu \gamma} \sum_{i=1}^{i=\mu} \gamma^{i/\mu} - 1 \right] - \tau_M \mu_2 \quad (7.9)$$

re constants computed once before implementation. Therefore

$$v_{pr}(k/k-1) = \bar{z}_{pr}(k) - \hat{x}_1(k/k-1) \quad (7.10)$$

We note that if $\mu = 1$, then μ_2 and μ_3 vanish and $z_{pf}(k)$ becomes $z(k)$ as before.

Let us now consider the effect of prefiltering on the measurement error statistics. From Equation (7.7), it is obvious that, strictly speaking, an error in z_{pf} is a function of the errors $\epsilon_2(k/k-1)$ and $\epsilon_3(k/k-1)$ as well as the measurement errors. However, consideration of the small values of μ_2 and μ_3 reveal these effects to usually be negligible. Under this assumption, we find that the prefiltered measurement error is simply the average of the individual measurement errors. That is

$$v_{pf}(k) = \frac{1}{\mu} \sum_{i=1}^{i=\mu} v(k-1+i/\mu) \quad (7.11)$$

The standard deviation of the prefiltered measurement error is (assuming the measurement error standard deviation constant)

$$\begin{aligned} \sigma_{pf}^2(k) &= E \{v_{pf}^2(k)\} \\ &= \frac{1}{\mu^2} E \left[\sum_{i=1}^{i=\mu} \sum_{j=1}^{j=\mu} v(k-1+i/\mu) v(k-1+j/\mu) \right] \end{aligned} \quad (7.12)$$

The term in the brackets is the average autocovariance of all the measurement error pairs. Each of these can be expressed in terms of the nominal autocorrelation coefficient for lag Δt as

$$\begin{aligned} R_{ij} &= E \{V(k-1+i/\mu) V(k-1+j/\mu)\} \\ &= \sigma^2(k) \exp \{-|t(i) - t(j)|/\tau\} \\ &= \sigma^2(k) \exp \left[-\frac{|i-j| \Delta t}{\mu \tau} \right] \\ &= \sigma^2(k) \rho(k)^{|i-j|/\mu} \end{aligned} \quad (7.13)$$

Substituting into Equation (7.12), we find

$$\sigma_{pf}^2(k)/\sigma^2(k) = \frac{1}{\mu^2} \sum_{i=1}^{i=\mu} \sum_{j=1}^{j=\mu} \rho(k)^{|i-j|/\mu} \quad (7.14)$$

Notice that, if $\rho(k)$ equals zero (white noise), and noting that 0^0 is unity, $\sigma_{pf}^2(k)/\sigma^2(k)$ is the familiar $1/\mu$. A similar expression can be written for the autocorrelation of the $V_{pf}(k)$'s which is

$$R_{pf}(k, k-1) = E[V_{pf}(k) V_{pf}(k-1)] \quad (7.15)$$

Using Equation (7.13), we find

$$R_{pf}(k, k-1)/\sigma^2(k) = \frac{1}{\mu^2} \sum_{i=1}^{i=\mu} \sum_{j=1}^{j=\mu} \rho(k)^{|i-j|+\mu/\mu} \quad (7.16)$$

In this case, if $\rho(k)$ is zero, $R_{pf}(k)$ is zero. Therefore, if the original measurement errors are uncorrelated, the prefiltered measurement errors will also be uncorrelated. Also, if $\mu = 1$, we recover $\sigma_{pf}^2(k) = \sigma^2(k) = \sigma^2(k)$ and $R_{pf}(k, k-1) = \sigma^2(k) \rho(k)$.

The correlation coefficient for the prefiltered measurement errors is

$$\rho_{pf}(k) = R_{pf}(k, k-1)/\sigma_{pf}^2(k) \quad (7.17)$$

and the effective prefiltered measurement correlation time constant can be calculated, if desired, as

$$\tau_{pf}(k) = \Delta t / \ln[1/\rho_{pf}(k)] \quad (7.18)$$

In Figure 7.2, the ratios of σ_{pf}/σ and τ_{pf}/τ are plotted as a function of μ (as though μ were a continuous variable). Since σ_{pf}/σ and τ_{pf}/τ are functions only of μ and the nominal correlation coefficient ρ , several values of τ were chosen to specify ρ . The nominal data rate is $k = 16$ Hertz so that the filter cycling time for each value of μ on the graphs is $\Delta t = \mu/k$. We find that the ratio σ_{pf}/σ is always < 1 for $\mu > 1$, and the measurement error reduction is greatest for the white noise case. The ratio of τ_{pf}/τ is > 1 and increases, on the other hand, as μ increases. The ratio decreases as τ increases, however.

Let us now consider the effect of prefiltering on filter performance. We find there are three factors whose interaction we must consider. The increase in the time increment between filter cycles (a reduction of the filter cycle rate) and the increase in the autocorrelation of the prefiltered error both tend to increase covariance and thus degrade performance. Fortunately, however, the decrease in the prefiltered measurement error variance tends to decrease the error covariance and improve filter performance. The net effect on filter performance was found to be essentially negligible with less than 2 percent variation in $\sigma_p(t + 10/t)$ when τ ranges from 0 to 0.5 seconds and μ varies from 1 to 10. It has therefore been concluded that this method of prefiltering, based upon residual averaging, offers us a

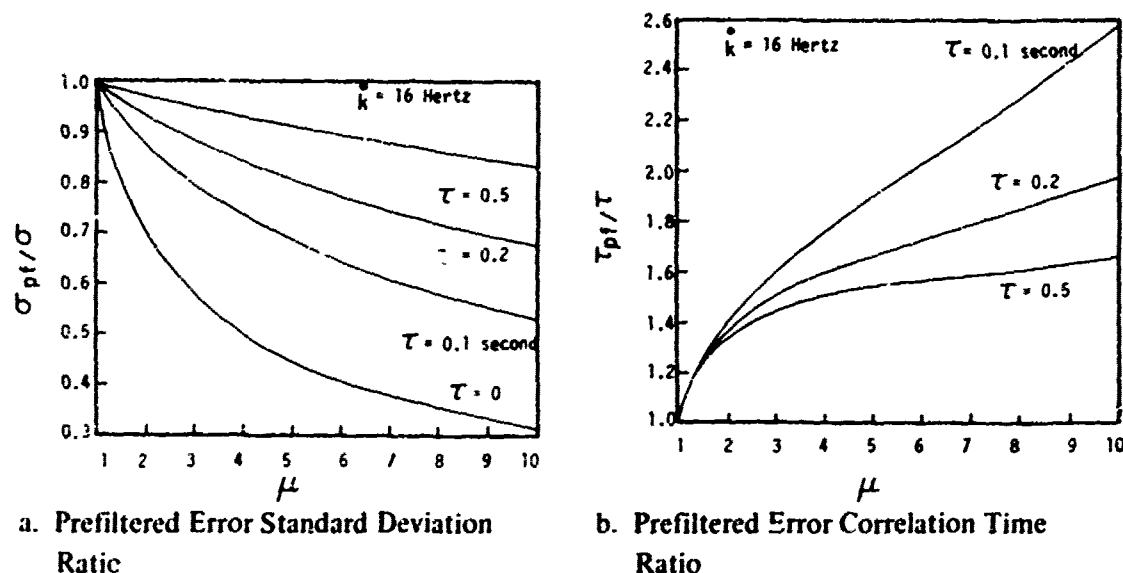


Figure 7.2

computationally inexpensive way to achieve high-data-rate filtering with negligible sacrifice in performance relative to high-data-rate full-Kalman filtering.

A small problem was encountered from the use of this method of prefiltering which will now be discussed along with the solution. As mentioned previously, the residual averaging technique of prefiltering causes the actual effective measurement error statistics to be a function of the estimation error, as can be observed in Equation (7.7). It was previously assumed that this portion of the prefiltered measurement error was negligible as far as the calculation of the effective prefilter statistics is concerned. This assumption is valid except during the initial coverage period when the estimation error covariance is quite large. Upon comparing the calculated covariance with the actual (simulated) Monte Carlo error, it was discovered that, during the initial coverage interval, the actual errors consistently exceeded the calculated error standard deviation but that the effect disappeared after a couple of seconds. The author refers to the phenomenon as "prefilter lag," which can be explained by considering Figure 7.3. During the initial convergence period, the expected value of the residuals evaluated at the times of actual measurement, as shown by the circles, can significantly increase over the prefiltering interval Δt . The lag problem occurs because an average of these residuals does not have the expected error level (variance) that the filter calculates. The standard deviation of the actual residual average is always less than the value which the filter expects, $\sigma_p/(k/k - 1)$.

After trying several methods to rectify this situation, the author finally found a technique whereby the prefiltered residual would be "scaled up" so that its variance would

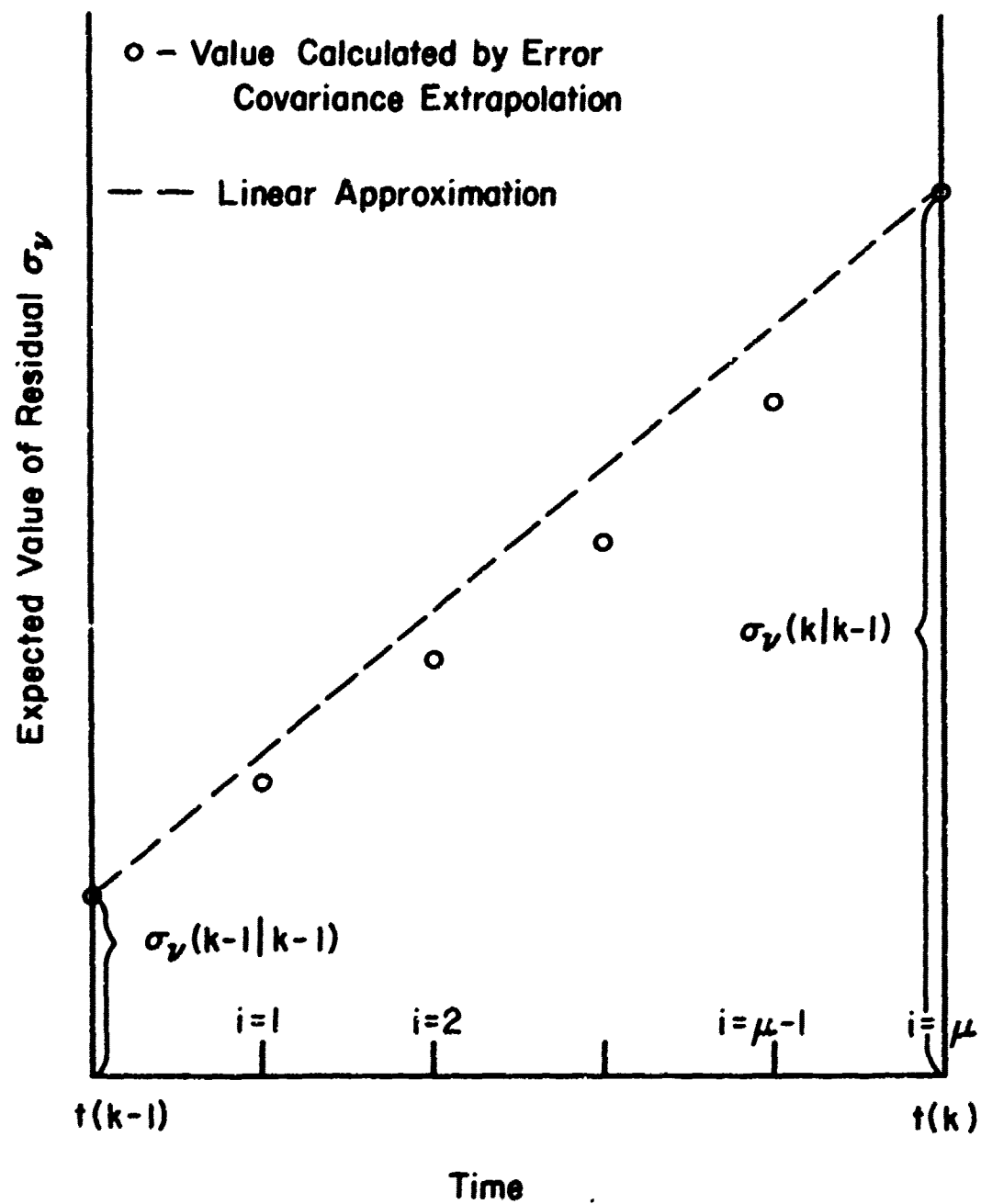


Figure 7.3. Conceptual Diagram of the "Prefilter Lag" Problem

match that calculated by the filter. This was done as follows. First it was noticed that the variation of σ_v over the interval Δt is approximately linear (since velocity estimation error dominates) so that we can write σ_v evaluated at the measurement points i as

$$\sigma_v(i) = (1 - i/\mu) \sigma_v(k - 1/k - 1) + (i/\mu) \sigma_v(k/k - 1) \quad (7.19)$$

The average value of $\sigma_v(i)$ occurs at the point $i = (\mu - 1)/2$ or

$$\sigma_{v\text{AVG}}(k) = \left(\frac{\mu + 1}{2\mu}\right) \sigma_v(k - 1/k - 1) + \left(\frac{\mu - 1}{2\mu}\right) \sigma_v(k/k - 1) \quad (7.20)$$

Finally, the factor by which we want to scale the prefiltered residual is $\sigma_v(k/k - 1)/\sigma_{v\text{AVG}}(k)$ or

$$f_{pt,r}(k) = \frac{1}{\left[\frac{1 + r_r(k)}{2}\right] + \left[\frac{r_r(k) - 1}{2\mu}\right]} \quad (7.21)$$

where

$$r_r(k) = \sigma_v(k - 1/k - 1)/\sigma_v(k/k - 1) \quad (7.22)$$

is the ratio of residual levels across the interval. Clearly, if there is no prefiltering ($\mu = 1$) or if the filter is in steady-state operation ($r \approx 1$), this factor is one and does not influence performance. It was observed, however, that this prefiltered residual scaling eliminated the prefilter entirely and was an economical "fix" to implement.

Concerning future work in the area of prefiltering, the author believes that an even greater amount of data compression can be achieved by using simple data processing techniques such as least squares. The author, as stated previously, feels that target motion is highly band-limited in that the upper limit of frequency is fairly discernible from aerodynamic limitations of aircraft and missiles. Consideration of the Shannon sampling theorem, with the usual factor of ten put in to account for noisy data, tells us that it is necessary to cycle our target motion estimator at only 2 to 1 Hertz. Actually, cycling at 1 Hertz is marginal, according to Shannon, but we are probably not interested in (or capable of) actually recovering all the very highest frequency target motion but merely want to track through it without diverging. Also, very-high-frequency target acceleration usually results in very small actual displacement that tends to be unobservable as it is down in the measurement noise. A simple, constant-velocity least-squares fit over, say, a compression interval of 1 second and a data rate of perhaps 32 Hertz appears very attractive at this time and should be performance-tested as soon as possible.

B. COVARIANCE SQUARE ROOT

The covariance square root method always insures a symmetric positive definite error covariance matrix. More significantly, the square root formulation can propagate the error covariance in single precision as accurately as the conventional error covariance methods do in double precision. Basically, this technique propagates a matrix S , instead of P , where S is defined by the relation

$$P(k/j) = S(k/j) S^T(k/j) \quad (7.23)$$

This definition of S is not unique, however. For example, in our situation, P is a (3×3) symmetric matrix specified by 6 variables. Since S is also a (3×3) matrix, for which 9 variables are required for specification, we find ourselves with three extra degrees of freedom in S which might be used advantageously. We have chosen, for reasons to be made clear shortly, to complete the definition by requiring S to be upper-triangular. As we shall find, the definition of S as upper-triangular does not insure that S will remain in this form when propagated through the filter equations. The upper-triangular definition of S requires

$$P_{11}(k/j) = S_{11}^2(k/j) + S_{12}^2(k/j) + S_{13}^2(k/j) \quad (7.24a)$$

$$P_{12}(k/j) = S_{12}(k/j) S_{22}(k/j) + S_{13}(k/j) S_{23}(k/j) \quad (7.24b)$$

$$P_{13}(k/j) = S_{13}(k/j) S_{33}(k/j) \quad (7.24c)$$

$$P_{22}(k/j) = S_{22}^2(k/j) + S_{23}^2(k/j) \quad (7.24d)$$

$$P_{23}(k/j) = S_{23}(k/j) S_{33}(k/j) \quad (7.24e)$$

$$P_{33}(k/j) = S_{33}^2(k/j) \quad (7.24f)$$

The inverse relations are

$$S_{33}(k/j) = \sqrt{P_{33}(k/j)} \quad (7.25a)$$

$$S_{23}(k/j) = P_{23}(k/j)/S_{33}(k/j) \quad (7.25b)$$

$$S_{22}(k/j) = \sqrt{P_{22}(k/j) - S_{23}^2(k/j)} \quad (7.25c)$$

$$S_{13}(k/j) = P_{13}(k/j)/S_{33}(k/j) \quad (7.25d)$$

$$S_{12}(k/j) = [P_{12}(k/j) - S_{13}(k/j) S_{23}(k/j)]/S_{22}(k/j) \quad (7.25e)$$

$$S_{11}(k/j) = \sqrt{P_{11}(k/j) - S_{12}^2(k/j) - S_{13}^2(k/j)} \quad (7.25f)$$

Now let us consider the filter covariance equations as given in Table 6.2. The error covariance extrapolation equation can be written as

$$P(k/k-1) = P'(k/k-1) + Q(k-1) \quad (7.26)$$

where

$$P'(k/k-1) = \phi(k, k-1) P(k-1/k-1) \phi^T(k, k-1) \quad (7.27)$$

It is easily seen that the extrapolation to $P'(k/k-1)$ can be equivalently accomplished by the square root as

$$S'(k/k-1) = \phi(k, k-1) S(k-1/k-1) \quad (7.28)$$

Applying the transition matrix for our case, we find that, if $S(k-1/k-1)$ is upper-triangular, then $S'(k/k-1)$ is also upper-triangular with the following values.

$$S'_{11}(k/k-1) = S_{11}(k-1/k-1) \quad (7.29a)$$

$$S'_{12}(k/k-1) = S_{12}(k-1/k-1) + \Delta t S_{22}(k-1/k-1) \quad (7.29b)$$

$$S'_{13}(k/k-1) = S_{13}(k-1/k-1) + \Delta t S_{23}(k-1/k-1) + \alpha S_{33}(k-1/k-1) \quad (7.29c)$$

$$S'_{22}(k/k-1) = S_{22}(k-1/k-1) \quad (7.29d)$$

$$S'_{23}(k/k-1) = S_{23}(k-1/k-1) + \beta S_{33}(k-1/k-1) \quad (7.29e)$$

$$S'_{33}(k/k-1) = \gamma S_{33}(k-1/k-1) \quad (7.29f)$$

If process noise, Q , is present, the addition of this term poses a more difficult problem in the square root covariance formulation. There are several possible methods, both exact and approximate, discussed in the literature. The simplest and usually the fastest of the exact methods is the so-called root-sum-square (RSS) operation discussed by Carlson (1973) among others. This technique essentially recalculates the error covariance, using Equation (7.24), adds the process noise and then takes the matrix square root to determine

$$S(k/k-1) = [S'(k/k-1) S'^T(k/k-1) + Q(k-1)]^{1/2} \quad (7.30)$$

This last operation, using Equation (7.25), is commonly known as Cholesky decomposition in triangular form. An alternative exact method is the Householder triangularization algorithm, discussed by Kaminski, Bryson and Schmidt (1971) which maintains double precision accuracy but with higher computational cost. Schmidt (1970) also discusses—but does not recommend—a technique which requires two matrix inversions and results in a loss of the upper triangular form. In addition to considering these exact algorithms, the author also found the technique of Wu (1973) very interesting. Wu assumed a square root additive process noise of a form

$$S(k/k-1) = S'(k/k-1) + q'(k-1) \quad (7.31)$$

where we might think of $q'(k-1)$ as equivalent in some sense to $Q^{1/2}(k-1)$. Equation (7.31) is admittedly not equivalent to the original extrapolation, Equation (7.26), since two additional terms appear in the error covariance. That is

$$\begin{aligned} P(k/k-1) = & S'(k/k-1) S'^T(k/k-1) + q'(k-1) q'^T(k-1) \\ & + S'(k/k-1) q'^T(k-1) + q'(k-1) S'^T(k/k-1) \end{aligned} \quad (7.32)$$

Wu's principal argument for this assumption is that the process noise itself (Q) is rarely known exactly in the first place, being "basically empirically determined data." The method is also no different, in principle, from the epsilon technique in which a somewhat arbitrary process noise is added to prevent divergence. Since divergence prevention and control of the filter bandwidth are the primary uses of process noise in our application, this argument does not seem unreasonable. The main difficulty in Wu's method is in application. It is very difficult to control the amount of process noise which we would like to add. Notice that the last two terms involve S' so that the amount of process noise actually added is a function of the current error covariance. These two terms are definitely not negligible since each is usually larger than the $q' q'^T$ term. The reason for this is that S' is usually much larger than q' . The author found that, by trial and error, it is possible to find a q' (much less than $Q^{1/2}$) that produces essentially the same steady-state error covariance as the covariance filter for a given situation. Unfortunately, no relation can be found between q' and Q which we might use to maintain adequate control over the filter bandwidth. Wu's method was therefore abandoned in favor of an exact method.

It was therefore decided to use the RSS technique which is the most efficient of the exact methods. The equations for adding process noise are then (letting $W = S(k/k-1)$ for notational convenience)

$$W_{33} = \sqrt{S'_{33} + Q_{33}} \quad (7.33a)$$

$$W_{23} = (S'_{23} S'_{33} + Q_{23})/W_{33} \quad (7.33b)$$

$$W_{22} = \sqrt{S'_{22} + S'_{23}^2 Q_{22} - W_{23}^2} \quad (7.33c)$$

$$W_{13} = (S'_{13} S'_{33} + Q_{13})/W_{33} \quad (7.33d)$$

$$W_{12} = (S'_{12} S'_{22} + S'_{13} S'_{23} + Q_{12} - W_{13} W_{23}/W_{22}) \quad (7.33e)$$

$$W_{11} = \sqrt{S'_{11} + S'_{12}^2 + S'_{13}^2 + Q_{11} - W_{12}^2 - W_{13}^2} \quad (7.33f)$$

Actually, it might appear—at least it did to the author—that the requirement to calculate the covariance in the RSS method would undermine the expressed desire to retain double-precision accuracy. The accuracy is not lost, however, since in the type of fixed-point computer in which this is implemented, a double-length word results automatically whenever two single length words are multiplied and a double length word must be used in order to obtain a single-length word from the square root. The net result is that, while double length words *appear* in the intermediate calculations, all the calculations are really in single precision and very efficient.

Let us now consider the update equation. The original technique for measurement update of the white noise filter was developed by Potter (1963) for the case of scalar measurements. Unfortunately, the Potter technique is not applicable to the form of the covariance update for the correlated measurement noise filter. The Potter form also results in a loss of triangularity even for the white noise case. We will therefore again utilize the RSS method suggested by Carlson in order to maintain double-precision accuracy. The covariance update equation, repeated from Table 6.2, is

$$P(k/k) = W W^T - K^*(k) G(k) K^{*T}(k)$$

Applying the modified RSS technique to our equations, we find (again letting $S = S(k/k)$ for convenience)

$$S_{33} = \sqrt{W_{33}^2 - G K_3^*{}^2} \quad (7.34a)$$

$$S_{23} = (W_{23} W_{33} - G K_2^* K_3^*)/S_{33} \quad (7.34b)$$

$$S_{22} = \sqrt{W_{22}^2 + W_{23}^2 - G K_2^*{}^2 - S_{23}^2} \quad (7.34c)$$

$$S_{13} = (W_{13} W_{33} - G K_1^* K_3^*)/S_{33} \quad (7.34d)$$

$$S_{12} = (W_{12} W_{22} + W_{13} W_{23} - G K_1^* K_2^* - S_{13} S_{23})/S_{22} \quad (7.34e)$$

$$S_{11} = \sqrt{W_{11}^2 + W_{12}^2 + W_{13}^2 - G K_1^{*2} - S_{12}^2 - S_{13}^2} \quad (7.34f)$$

Equations (7.33) and (7.34) will now replace the covariance equations (8) through (13) and (22) through (27) of Table 6.3. The gains are calculated in the same manner except that Equation (7.24) must be used to calculate the required covariances. The expressions for the A's are somewhat simpler in square root form, i.e.,

$$\begin{aligned} A_1(k) = \rho(k) [S_{11}'(k/k-1) + S_{12}'(k/k-1) S_{12}(k-1/k-1) \\ + S_{13}'(k/k-1) S_{13}(k-1/k-1)] \end{aligned} \quad (7.35a)$$

$$A_2(k) = \rho(k) [S_{12}(k-1/k-1) S_{22}'(k/k-1) + S_{13}(k-1/k-1) S_{23}'(k/k-1)] \quad (7.35b)$$

$$A_3(k) = \rho(k) S_{13}(k-1/k-1) S_{33}'(k/k-1) \quad (7.35c)$$

Initialization of the square root covariance is accomplished by another RSS, similar to Equation (7.25).

In summary, we have found it is possible to reduce the computational burden by a factor of μ by prefiltering without loss of performance. Typically, we might choose $\mu = 4$ and thereby process data at a rate of 16 Hertz but only cycle the filter at 4 Hertz. By using the square-root covariance filter, it was possible to eliminate double-precision covariance calculation, thereby significantly reducing the time required for covariance computation. In fact, a comparison was performed for the Navy standard mini-computer, the 16-bit A/N UYK 20, to estimate the difference between the error covariance equations implemented in double precision and the square-root covariance equations in single precision. The difference was more dramatic than anticipated as the square-root covariance resulted in a reduction of computer time by a factor of 4.56! This will translate into an overall filter computation reduction of a factor slightly less than four—since the filter obviously does other things besides covariance calculations. The combination of both these techniques reduced the required filter computation time by a factor of 12 to 16 over the high-data-rate covariance filter. We find that these reductions are well advised since the dual-bandwidth adaptive filter introduces an increase factor of two and the three-dimensional filter (Section IX) an increase factor of three. That is, we will be (approximately) able to run a three-dimensional, double adaptive filter with prefiltering and square root covariance in one half the time of single one-dimensional error-covariance filter without prefiltering.

VIII. PREDICTION*

This section concerns itself with the question of the prediction of future target position based on current estimates of position and rates. We will, for the moment, divorce ourselves from the target models utilized for the filter and reconsider models for the long-time prediction problem (i.e., $t_p = 5$ to 30 seconds). We will think of the filter only as a "black-box" algorithm which provides as outputs estimates of current position, velocity, acceleration, and associated error covariance. This information will serve as input to the predictor. The question is how the predictor can best use this information. The predictor, of course, is the critical factor in a gunfire control system in that the calculated future target position is the aim point for the gun.

The conventional method of prediction is to simply extrapolate the assumed dynamics model used in the filter itself. For example, a second-order derivative polynomial filter (with appropriate divergence prevention modifications) that provides estimates of the (possibly varying) acceleration would predict the position at time t_p in the future as

$$\hat{x}_A(t + t_p/t) = \hat{x}_1(t/t) + \hat{x}_2(t/t) t_p + \hat{x}_3(t/t) t_p^2/2 \quad (8.1)$$

where $\hat{x}(t/t)$ is the current state vector estimate. It has been suggested by some people that perhaps the estimated acceleration should not be used in the prediction even though such an estimate might be available from the filter. This argument is based on the indisputable contention that one cannot expect a target to maintain a constant acceleration over long prediction time intervals. Obviously if the acceleration is ignored in the predictor, we simply predict tangent to the curve and have the constant velocity predictor.

$$\hat{x}_v(t + t_p/t) = \hat{x}_1(t/t) + \hat{x}_2(t/t) t_p \quad (8.2)$$

This predictor is currently more prevalent in operational gunfire control systems.

This technique, of course, essentially surrenders any hope of, at least partially, dealing with an accelerating target. One is therefore faced with a choice of predictors, and it must be conceded at this point that, in the general case, one has no way of knowing what the target will do in the future. It might very well follow either Equation (8.1) or (8.2) or (more likely) neither. The prediction time is an important consideration here. For short prediction times (say 1 to 2 seconds), it does not make a great deal of difference since very few targets can maneuver at a level to seriously alter their course in such a short time. For longer prediction times (say 2 to 20 seconds), typifying the possible air target projectile flight times observed in large caliber gun systems, it is well known that an air target can alter its course

*This section resulted from discussions of the author with Dr. Charles J. Cohen and Mr. Tom Alexander of NSWC to whom the author is indebted for suggestions.

considerably. Indeed, it is the author's contention that, in an engagement, one should expect the target *not* to maintain a constant velocity or acceleration over such durations.

The above reasoning led rather naturally to a consideration of methods of somehow weighting the acceleration in the prediction equation. The weighting parameters required could conceivably be estimated by considering the types of target trajectories in a system scenario to determine parametric values that could improve prediction accuracy. One approach to this problem can be effected if we again consider the target's acceleration to be modeled as an exponentially autocorrelated random variable (as in Section IV) and therefore of a nondeterministic nature. Such a model would conceivably contain any target maneuver, be it the exercise of deterministic strategies, evasive maneuvers, acceleration perturbations due to winds and turbulence, etc., as simply a sample trajectory as long as the trajectory acceleration profile can reasonably be described by the chosen statistics.

Let us first consider the prediction transition matrix. If we again assume that the actual state obeys the relation

$$\underline{x}(t + t_p) = \Phi(t_p) \underline{x}(t) + \underline{w}(t + t_p, t) \quad (8.3)$$

then, if we are given an estimate $\hat{\underline{x}}(t)$, the expected value of $\hat{\underline{x}}(t + t_p)$ is simply

$$\hat{\underline{x}}(t + t_p/t) = \Phi(t_p) \hat{\underline{x}}(t/t) \quad (8.4)$$

since w is a zero mean random variable. Notice that we have used additional notation on w to indicate that it is integrated over t_p . It is quite easy to confirm that the transition matrix also yields the minimum variance linear prediction. Suppose we assume some transition matrix (say A) which we want to determine so as to yield the minimum prediction error variance, i.e., we assume

$$\hat{\underline{x}}(t + t_p/t) = A(t_p) \hat{\underline{x}}(t/t) \quad (8.5)$$

The prediction error is then

$$\begin{aligned} \underline{\epsilon}(t + t_p/t) &= \hat{\underline{x}}(t + t_p/t) - \underline{x}(t + t_p) \\ &= A(t_p) \hat{\underline{x}}(t/t) - \Phi(t_p) \underline{x}(t) - \underline{w}(t + t_p, t) \end{aligned} \quad (8.6)$$

and the prediction error covariance is

$$\begin{aligned}
P(t + t_p | t) &= E [\underline{\epsilon}(t + t_p | t) \underline{\epsilon}^T(t + t_p | t)] \\
&= [A(t_p) - \Phi(t_p)] \Omega(0) [A(t_p) - \Phi(t_p)]^T \\
&\quad + A(t_p) P(t | t) A^T(t_p) + Q(t_p)
\end{aligned} \tag{8.7}$$

The matrix $\Omega(0)$ is the covariance of \underline{x} where

$$\Omega(T) = E [\underline{x}(t) \underline{x}^T(t + T)]$$

is the autocovariance of \underline{x} which can be related to the transition and process noise matrices by taking moments of Equation (8.3), i.e.,

$$\Omega(0) - \Phi(t_p) \Omega(0) \Phi^T(t_p) = Q(t_p) \tag{8.8}$$

and

$$\Omega(t_p) = \Phi(t_p) \Omega(0) \tag{8.9}$$

Using the techniques described in Gelb (1974), we construct the cost function

$$J = E [\epsilon^T(t + t_p/t) S \epsilon(t + t_p/t)] \tag{8.10}$$

where S is any positive semidefinite matrix. Choosing $S = I$, we find

$$J = \text{trace} [P(t + t_p/t)] \tag{8.11}$$

It is a relatively simple exercise to confirm that the necessary and sufficient condition to minimize J is to choose

$$A(t_p) = \Phi(t_p) \tag{8.12}$$

We find, therefore, that the minimum variance predictor is also given by the transition matrix. The position prediction equation is then

$$\begin{aligned}
\hat{x}_1(t + t_p/t) &= \hat{x}_1(t/t) + \hat{x}_2(t/t) t_p \\
&\quad + \hat{x}_3(t/t) \tau_p^2 [\exp(-t_p/\tau_p) + t_p/\tau_p - 1]
\end{aligned} \tag{8.13}$$

which is the double integral of the prediction acceleration

$$\hat{x}_3(t + t_p/t) = \hat{x}_3(t/t) \exp(-t_p/\tau_p) \tag{8.14}$$

Notice that we have introduced another parameter, τ_p , which is the acceleration time constant we want to use for prediction. This value does not necessarily correspond to any previous parameters since, as mentioned in previous sections, the filter does not assume any particular value of the maneuver frequency but instead assumes an upper and lower bound of a range of values. The author, at one time, attempted to determine a value of τ_p on line (i.e., during real-time execution) or adaptively, but to date no satisfactory technique has been found. It still appears feasible, however, to estimate an appropriate value of τ_p based upon the state of convergence of the adaptive filter and additional work in this area might be productive.

In the absence of such a technique, however, it appeared more promising to use an expected value

$$\tau_p = E[\tau_M] \quad (8.15)$$

where the expectation is taken over the target scenario. At this time, a value of $\tau_p = 5$ seconds appears most reasonable to the author. It should be noted that the actual value of the predictor is rather sensitive to the choice of τ_p , as can be seen in Figure 8.1. The acceleration at $t = 15$ seconds is assumed to be known as 4 yards/second². By choosing different values of τ_p (from zero to infinity), we find that the actual predicted position (say for $t_p = 15$ seconds) varies by several hundred yards. If τ_p is zero, we essentially are using constant-velocity prediction, and conversely, if τ_p becomes very large (or frequency approaches zero), we predict a constant acceleration path. For values in between zero and infinity, the predicted acceleration decays exponentially, and the path eventually approaches a constant velocity—but not the same path as if $\tau_p = 0$. Notice that the exponential acceleration equation always predicts a path somewhere between constant velocity and constant acceleration and can never cross these bounds.

A study conducted by the author, Clark (1973), compared the three predictors (Equations 8.1, 8.2, and 8.13) discussed in this section. The results indicated that, while for any given trajectory, either constant velocity or constant acceleration might be the best, the exponential acceleration predictor was consistently the overall best (considering root-sum-square prediction error). This result should be expected since the exponential acceleration matches the acceleration autocovariance of the scenario much better than the other two predictors.

Before continuing the filter development, it is interesting—and very disheartening—to consider realistically the type of prediction error likely to be experienced by a gunfire control system against a “randomly” maneuvering target under conditions of perfect filtering (where we assume the current filter error covariance vanishes). The standard deviation

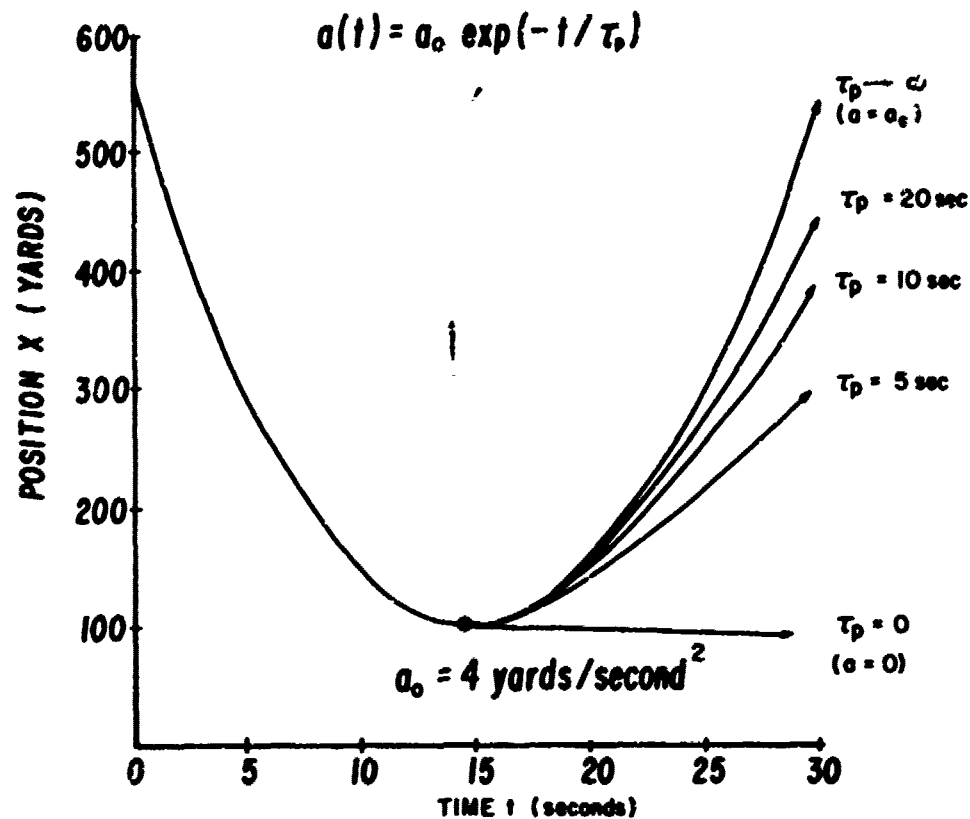


Figure 8.1. Random Acceleration (Exponential) Prediction: Examples

of the predicted position error is due only to extrapolated process noise. That is

$$\sigma_1(t + t_p/t) = \sqrt{Q_{11p}} \quad (8.16)$$

where Q_{11p} is calculated from Equation (4.16). In Figure 8.2, this quantity is plotted as a function of prediction time t_p for a low, medium, and high set of the parameters σ_m and τ_M . These results are consistent with those obtained by simulation against targets with the same maneuvering characteristics. When we consider these values in light of the effective radii of lethality of various projectile/target combinations, we must conclude that the effectiveness of such gun systems (using unguided projectiles) at long ranges is certainly questionable. In fact, this unfortunate situation is the primary factor limiting the effectiveness of current conventional gunfire control systems.

The eventual deployment of a terminally guided projectile with significantly larger acquisition "baskets" promises a corresponding increase in the effectiveness of gun systems

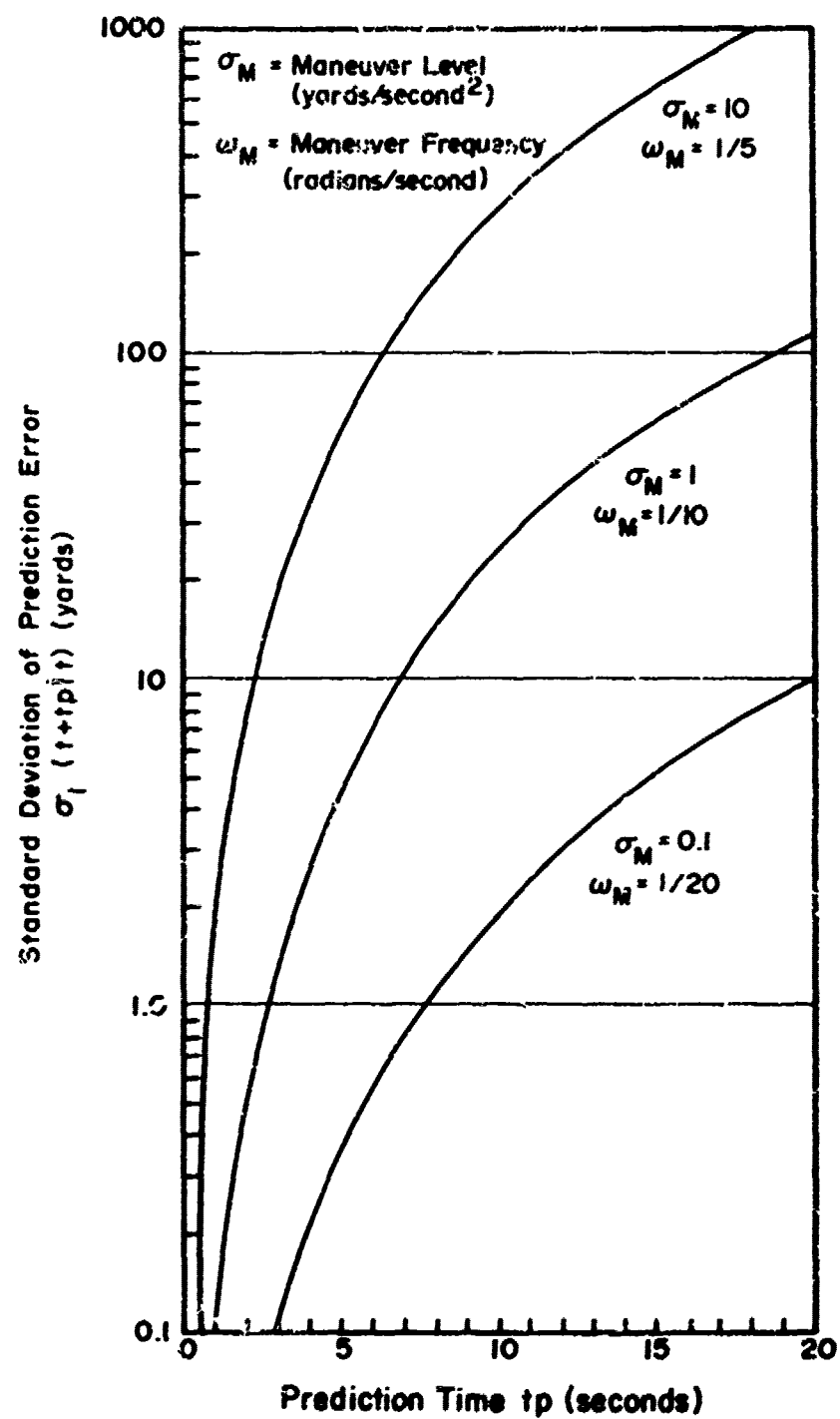


Figure 8.2. Prediction Errors for Random Acceleration Target Assuming Perfect Filter (Zero Filter Error Covariance)

against maneuvering targets. The random acceleration model, as developed in this paper, is not intended to be a "solution" to the gunfire control prediction problem. It is really only a more realistic approach to the problem, as opposed to the derivative polynomial models, and it is felt an improvement. The author and his associates are currently investigating new prediction techniques that hold the promise of significantly improved prediction accuracies for particular targets. A report on these techniques is expected to be published in the near future.

IX. THREE-DIMENSIONAL CARTESIAN FILTERING

All discussion to this point on filtering and prediction have dealt with one-dimensional estimation of target motion. The rationale for this approach was to maintain a level of simplicity in the presentation as long as possible. Obviously, the multidimensional aspects of the problem actually had to be considered at all times during the development. In this section, we will concentrate our attention on this subject.

Several factors relating to the application must be considered in selecting a coordinate system for filtering and prediction. Primary consideration must be given to accuracy, since this is a fire control problem, and computational simplicity, since the algorithm must be implemented within the constraints of a given computer. The sensitivity to various non-linear effects, we shall find, drives the selection based upon these factors. Special types of measurements (such as Doppler range rate) or the existence of multiple rates between the polar measurements would influence this decision but are not present for this application. A factor which is important in this instance is the reference coordinate system required for actual filter input and predictor output. The requirement for stabilized (pitch and roll corrected) measurements with a common reference point on output can be effected with fewer and simpler transformations in the Cartesian frame, but this factor is not of primary interest. The first two factors, accuracy and computational burden, were therefore the driving forces for this study and will therefore be considered in depth in this section.

The study presented in this section is based on two premises:

(1) Measurements of target position are obtained in spherical polar coordinates, i.e., slant range (R), bearing angle from north (B) and elevation angle (E). It is assumed that, as designer of a filter/predictor for a given fire control system, we have available "reasonably good" estimates of the quality of these measurements, i.e., measurement error variances and correlations in the original (polar) measurement frame. It is also assumed that the designer knows the availability of this data such as the range of possible data rates, the volumetric coverage of the sensor, any spatial variation of the measurement error statistics (due to such effects as multipath), and estimates of any target induced errors such as glint and scintillation.

The polar-Cartesian transformation equations are:

$$\underline{x}_C = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \sin B \cos E \\ r \cos B \cos E \\ r \sin E \end{bmatrix} = \underline{x}_C(\underline{x}_P) \quad (9.1)$$

where x is east, y is north and z is vertical. The corresponding inverse relation is:

$$\underline{x}_P = \begin{bmatrix} r \\ B \\ E \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ \arctan(x/y) \\ \arcsin(z/r) \end{bmatrix} = \underline{x}_P(\underline{x}_C) \quad (9.2)$$

(2) The target is *assumed* to be modeled "best" in Cartesian coordinates. By this, we mean that the target is more closely linear and well behaved in Cartesian coordinates than in polar coordinates. For example, let us consider simple linear target motion which is canonical in Cartesian coordinates.

$$dx/dt = v_x = \text{Constant} \quad (9.3a)$$

$$dy/dt = v_y = \text{Constant} \quad (9.3b)$$

$$z = 0 = \text{Constant} \quad (9.3c)$$

Using Equation (9.2), we find that second (and all higher) derivatives appear in the polar frame as

$$d^2 r/dt^2 = r (dB/dt)^2 \quad (9.4a)$$

$$d^2 B/dt^2 = -2 (dB/dt)(dr/dt)/r \quad (9.4b)$$

$$dB/dt = (x \cdot v_x - y \cdot v_y)/r^2 \quad (9.4c)$$

The author refers to these second derivatives as "apparent accelerations" and they are described in the literature, such as Monzingo (1972) who calls them "psuedo-maneuvers" and Cantrell (1973). These accelerations, if viewed in the polar frame, must either be modeled and propagated nonlinearly by the filter/predictor or, even worse, tracked adaptively. Notice also, in Equation (9.4), that any crossing component can produce very high angular rates if r becomes small. There are, of course, corresponding polar canonical targets

such as motion along a ray or circular motion centered at the origin. Examination of the target scenario indicates, however, that such motion would be encountered much less frequently than the nominal Cartesian motion.

Perhaps the "best" target coordinate system of all would be one translating with the target and oriented appropriately along the (changing) velocity vector and two normal acceleration directions. Such a coordinate system would be very cumbersome to implement, however, and would not change the fact that a basic nonlinearity exists between the measurement frame and the target motion frame. We will therefore assume that the Cartesian random acceleration target model, where the maneuver level and frequency lie within the previously described bounds, represents the true target and proceed on that basis. To form the Cartesian target model, an additional assumption of independence of target maneuver between channels was made. That is, we define the three-dimensional Cartesian state vector

$$\underline{x}_{3DC} = [\underline{x} \ \underline{y} \ \underline{z}]^T \quad (9.5)$$

where \underline{x} , \underline{y} , and \underline{z} are each one-dimensional three-element state vectors (position, velocity, and acceleration) governed by the random acceleration model of Section IV. The transition matrix is then

$$\Phi_{3DC} = \begin{bmatrix} \phi & 0 & 0 \\ 0 & \phi & 0 \\ 0 & 0 & \phi \end{bmatrix} \quad (9.6)$$

where ϕ is defined by Equation (4.8) and the process noise is

$$Q_{3DC} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix} \quad (9.7)$$

where Q is given by Equation (4.13). The assumption of maneuver coordinate independence is weak but depends upon the particular type of target and its particular angular orientation in the coordinate system. The problem can probably be resolved by modeling the target in the moving target frame (mentioned previously) or by cross-correlation studies. Clearly, the target motion analysis problem is one that requires much additional attention.

Starting with premises (1) and (2), it becomes immediately obvious that there are a startling number of different filter configurations, both optimal and suboptimal, that one

can construct. Primary classification of these configurations will be by coordinate system of the state vector. We then considered three classes: (A) Cartesian; (B) spherical polar; and (C) hybrid (a combination of both). The details will be presented here only for the Cartesian portion of the study since a suboptimal version of this class was ultimately chosen. The performance criterion was again chosen to be the average (integrated) predicted position optimality ratio

$$\theta(t_p, T) = \frac{1}{kT} \sum_{k=1}^{k=KT} \frac{\sigma_{3D \text{ Optimal}}(t_k + t_p/t_k)}{\sigma_{3D \text{ Actual}}(t_k + t_p/t_k)} \quad (9.8)$$

where σ_{3D} is the root-sum-square of the three predicted position error coordinate values. Based on the premises of this study, we will now proceed to form filter configurations and test the performance of each.

A. OPTIMAL NONLINEAR FILTER

Given the Cartesian target model which is obviously linear in the Cartesian frame, the use of polar measurements implies that the measurement update step will contain the non-linearity. Now we will model this nonlinear measurement and construct an optimal nonlinear filter. The three-dimensional polar measurement is

$$\begin{aligned} \underline{z}_p &= [r \ B \ E]^T + v_p \\ &= \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ \arctan(x/y) \\ \arcsin(z/r) \end{bmatrix} + [v_r \ v_B \ v_E]^T \end{aligned} \quad (9.9)$$

with measurement error variance

$$R_p = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_B^2 & 0 \\ 0 & 0 & \sigma_E^2 \end{bmatrix} \quad (9.10)$$

assuming no cross correlation of polar errors. Equation (9.9) is clearly of the general nonlinear form

$$z(k) = h(\underline{x}) + \underline{v}(k) \quad (9.11)$$

which can be treated by using the "iterated extended Kalman filter." The extended Kalman filter linearizes $h(\underline{x})$ about the extrapolated estimate and uses the Jacobian to obtain the filter update. The iterated filter simply repeats the process each time, linearizing about the most recent estimate. Details of this general technique will not be presented in this report but can be found in Jazwinski (1970). An important point is that the iterated extended Kalman filter, if nondivergent, tends to remove those estimation errors which are caused by systematic and/or observation nonlinearities. By iterating until the changes between successive estimates and covariances become arbitrarily small, one approaches, as closely as desirable, the optimal estimator for the nonlinear situation. This is the technique utilized in this study to establish the optimal or standard against which we will compare various admittedly suboptimal filters.

B. COUPLED LINEARIZED FILTER

Another approach for three-dimensional Cartesian filtering involves utilizing the polar/Cartesian transformation to nonlinearly combine the polar observations to form Cartesian measurements. The three-dimensional measurements are given as

$$\underline{z}_C = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \sin B \cos E \\ r \cos B \cos E \\ r \sin E \end{bmatrix} + \underline{v}_C \quad (9.12)$$

with observation matrix

$$H_C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (9.13)$$

A problem arises since, although we input the exact nonlinear measurement, it is difficult or impossible to compute the exact nonlinear measurement error variance. In order to estimate R corresponding to Equation (9.12), we resort to linearizing the measurement error. The sensitivity to the approximation error of R should not be high as we found in the measurement error sensitivity curves in Figure 6.3(b). To form this linear approximation, we first must linearize the measurement error. Differentiating Equation (9.12) and replacing

the differentials by "deltas" representing the finite observation errors, we find that the linearized Cartesian errors are equivalent to a simple rotation of the polar errors.

$$\underline{v}_{CL} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = T(\underline{x}) \underline{v}_p \quad (9.14)$$

where the Jacobian is

$$T(\underline{x}) = \begin{bmatrix} sBcE & rcBcE & -rsBsE \\ cBcE & -rsBcE & -rcBsE \\ sE & 0 & 0 \end{bmatrix} \quad (9.15)$$

using s and c to represent \sin and \cosine .

Again assuming no cross correlation of polar errors, the linearized Cartesian measurement error variance is

$$R_{CL} = T(\underline{x}) R_p T^T(\underline{x}) \quad (9.16)$$

For reference purposes, the elements of R_{CL} will be written out.

$$\sigma_x^2 = sB^2cE^2 \sigma_r^2 + r^2cB^2 \sigma_\theta^2 + r^2sB^2sE^2 \sigma_E^2 \quad (9.17a)$$

$$\sigma_y^2 = cB^2cE^2 \sigma_r^2 + r^2sB^2 \sigma_\theta^2 + r^2cB^2sE^2 \sigma_E^2 \quad (9.17b)$$

$$\sigma_z^2 = sE^2 \sigma_r^2 + r^2cE^2 \sigma_E^2 \quad (9.17c)$$

$$\sigma_{xx} = \sigma_{yy} = sBcB(\sigma_r^2 cE^2 - r^2 \sigma_\theta^2 + r^2sE^2 \sigma_E^2) \quad (9.17d)$$

$$\sigma_{xz} = \sigma_{yz} = sBsEcE(\sigma_r^2 - r^2 \sigma_E^2) \quad (9.17e)$$

$$\sigma_{yy} = \sigma_{zz} = cBsEcE(\sigma_r^2 - r^2 \sigma_E^2) \quad (9.17f)$$

It should be noted that there is a slight discrepancy between Equations (9.16) and (9.17). Everywhere in Equation (9.17) that σ_θ^2 appears, there originally was $\sigma_\theta^2 \cos^2 E$ as calculated

from Equation (9.16). The reason the $\cos^2 E$ was omitted is that the quantity $\sigma_B^2 \cos^2 E$ actually is the value that remains constant in the tracking system over the range of elevation angles from 0 to 90 degrees. The angular resolution is really constant in the *dish* frame, i.e., in the polar coordinate system that is aligned with the sensor axis. As one traverses from zero elevation to 90 degrees, the "closing" of the fixed polar frame at the pole singularity causes a fixed error in dish frame to subtend a greater bearing angle. This behavior goes as the inverse cosine of elevation. All we have done then is to substitute for σ_B^2 the value $\sigma_{BD}^2 / \cos^2 E$ (subscript D stands for dish); the inverse $\cos^2 E$ cancels with the $\cos^2 E$ that appears in each equation with σ_B^2 and the subscript D is emitted and Equations (9.17) are obtained.

While maintaining that the sensitivity analysis provides the ultimate test of the suboptimality of a particular filter configuration, it was also felt that the validity of the error linearization in Equations (9.14) and (9.16) were central to the problem and that further study of this approximation would shed light on the sensitivity results. With this in mind, two simple tests related to the linearization approximation were conducted. First, it was desired to directly test the actual assumption of error linearity, i.e., Equation (9.14). This can easily be done by calculating the true nonlinear Cartesian error as

$$\underline{y}_C = \underline{x}_C(\underline{x}_P + \underline{y}_P) - \underline{x}_C(\underline{x}_P) \quad (9.18)$$

where the function \underline{x}_C is given by Equation (9.1). The error due to nonlinearity is then simply the difference

$$\underline{\eta}_C = \underline{y}_{CL} - \underline{y}_C \quad (9.19)$$

Actual values of $\underline{\eta}_C$ were then calculated over the range of coordinates of interest for the GFC

$$\begin{bmatrix} 0 \text{ yards} \\ 0 \text{ rad} \\ 0 \text{ rad} \end{bmatrix} \leq \underline{x}_P = \begin{bmatrix} r \\ B \\ E \end{bmatrix} \leq \begin{bmatrix} 35000 \text{ yards} \\ 2\pi \text{ rad} \\ \pi/2 \end{bmatrix} \quad (9.20)$$

and, more importantly, over the range of polar noise

$$\begin{bmatrix} 0 \text{ yards} \\ 0 \text{ mrad} \\ 0 \text{ mrad} \end{bmatrix} \leq \underline{v}_P = \begin{bmatrix} v_r \\ v_B \\ v_E \end{bmatrix} \leq \begin{bmatrix} 40 \text{ yards} \\ 8 \text{ mrad} \\ 8 \text{ mrad} \end{bmatrix} \quad (9.21)$$

which exceeds by perhaps an order of magnitude the level of error anticipated with the actual tracking sensors. For the conditions of Equations (9.20) and (9.21), the maximum difference between the linear error and the nonlinear error never exceeded 1.5 yards, i.e., $|m| \leq 1.5$ yards with the maximum occurring at a range of 35000 yards. It was concluded that, whereas the linearization is used only for the specification of the measurement error statistics and not for the observations themselves, the effect of this approximation on system performance would be negligible since (as we observed in a previous section) the sensitivity to utilizing the wrong error statistics is minimal. It must also be kept in mind that we can not really expect to be able to estimate the sensor polar statistics with nearly as much accuracy in any case.

Another interesting test was conducted to evaluate the effect of the nonlinearity on the transformed Cartesian distribution functions. In this test, at positions over the range of coordinates of (9.20), one thousand points each of polar noise were generated and the corresponding nonlinear noise values \underline{v}_C calculated by Equation (9.18). The chi-squared test was utilized to evaluate the normality of the errors. Chi-squared was first calculated for the original polar noise \underline{v}_P (the values summed for the three coordinates) and then for the corresponding sequence of \underline{v}_C . The results were at first very surprising since for every noise sequence, chi-squared was less for the nonlinear sequence \underline{v}_C than for the original polar sequence \underline{v}_P , indicating the nonlinearly transformed Cartesian noise was more nearly Gaussian than the original polar noise. In retrospect, however, the reason for this is quite simple in that, to the extent the linearity of Equation (9.14) is valid, the Cartesian errors are simple linear combinations of the polar errors. Of course, linear transformations of Gaussian distributions are also Gaussian (see for example, page 94 of Bendat and Piersol (1971)). We would therefore expect by the Central Limit Theorem (*on cit*) that the Cartesian errors, being approximately linear sums of the polar Gaussian random errors, would display a tendency to be Gaussian also. Indeed, this is what we observe from the chi-squared tests. This further reinforces the author's contention that the linearization provides an excellent approximation, and given that the input polar observation errors are Gaussian the corresponding Cartesian errors should also be Gaussian. (Note that it is possible to analytically describe the true nonlinear Cartesian distribution function given polar Gaussian noise by using the Jacobian of the transformation. Unfortunately, the author was unable to simplify the result into a form that is recognizable for purposes of interpretation.)

C. UNCOUPLED LINEARIZED FILTER

The two Cartesian filters presented so far have resulted in ninth-order state and covariance systems. Upon examination of the previous filter equations, it becomes immediately obvious that the only terms which couple the three directions are the off-diagonal term of

the R_{CL} matrix since the target dynamics were assumed uncoupled. If the off-diagonal terms of R_{CL} are assumed to be negligible (an assumption to be tested shortly), then the system can be represented as three third-order systems (or "channels" as we call them). This results in an enormous reduction in the computational burden if it can be accomplished without serious degradation of performance. In fact, the computer time utilized on the CDC 6700 for each implementation was monitored and the uncoupled filter found to be *less than one twentieth* the other two.

Again, it is interesting to directly test the validity of this assumption so that we might better understand the results of the sensitivity analysis to be presented. A very brief consideration of the off-diagonal measurement covariances, Equations (9.17d-f), is very instructive. For example, each cross term is weighted by a factor of form

$$\begin{aligned} a &= \sin \lambda \cos \lambda \\ &= 0.5 \sin 2\lambda \end{aligned}$$

It is very easy to see that the maximum value of a is 0.5 and its average magnitude or root-mean-square is only 0.125. Since the remaining factors are no greater than the diagonal variances, this means that the cross-correlation coefficients will not exceed these values. Further examination of the terms in parentheses is even more revealing. Notice that these terms always algebraically subtract, thus tending to further reduce the correlation coefficients. In fact, for typical values of polar sensor statistics, the cross covariances all completely vanish at a range

$$r_s = \sigma_r / \sigma_\lambda$$

where σ_λ is the angular (bearing or elevation) standard derivation. Depending upon the particular sensor (and other conditions), r_s should usually be between 1000 and 12000 yards, with approximately 5000 yards most likely. Clearly, this range of r_s coincides with the tracking ranges of interest for Naval gunfire control. For all these reasons, we would expect the cross correlation of channels to be very weak and the suboptimal uncoupled filter to be close to optimal in performance.

D. SENSITIVITY: TEST CONDITIONS AND RESULTS

In order to evaluate the sensitivity of the two suboptimal filter configurations, a set of test conditions specifying polar track sensor statistics and targets must be chosen. Two hypothetical track sensors were parametrically determined in Table 9.1A. An older "sloppier" radar called Sensor "A" tracks to only 3 milliradians and suffers from serial correlation of bandwidth 1.88 Hertz due to the absence of off-axis corrected tracking. A newer

Table 9.1A. Track Sensor Statistics

Sensor A	=	Sensor E
5.0	σ_r (yards)	5.0
3.0	σ_B and σ_E (milliradians)	1.0
0.0	τ_r (seconds)	0.0
0.3	τ_B and τ_E (seconds)	0.0

Table 9.1B. Target Characteristics

MANEUVER STATISTICS

Maneuver Frequency	$\omega_M = 1/20 \text{ seconds}^{-1}$
Nonmaneuvering Target	$\sigma_{MA} = 0.1 \text{ yards/second}^2$
Maneuvering Target "B"	$\sigma_{MB} = 5.0 \text{ yards/second}^2$

NOMINAL TRAJECTORY

Closing Target	$r = 17500 \text{ to } 0 \text{ yards}$
Crossing Target	$X = \text{Constant} = 2500 \text{ yards}$
	$Y = -7500 + 300 \cdot t$

VELOCITY

Mach Number = 1.0

TARGET FLIGHT TIME

$T = 50 \text{ seconds}$

"tighter" radar called Sensor "E" adds one milliradian of white noise. Both sensors track with 5 yards of white noise in range. The targets are specified by choosing random maneuvers with given statistics superimposed upon a nominal straight-line target with given speed and orientation with respect to the ship. These conditions are presented in Table 9.1B. Two levels of σ_M were chosen to represent both maneuvering and nonmaneuvering targets. Two nominal transonic target paths were chosen: one closing directly from an initial range of 17500 yards; and one that crosses with a point of closest approach of 2500 yards. Fifty seconds of target track data were assumed to be supplied for estimation and prediction purposes.

The sensitivity test results shown in Table 9.2 are given in terms of the prediction criterion of Equation (9.8). First, consider the performance of the coupled linearized filter relative to the optimal nonlinear filter. In Section 2.B., we discussed and presented some auxiliary results which indicated that the linearization approximation was quite accurate for our application. If this linearization were perfect, the linearized coupled filter would be optimal. We find that, indeed, using our criterion of optimality, there is essentially no difference in the performance level of the coupled linearized filter relative to that of the optimal nonlinear filter. We also note that the performance is a strong function of the sensor statistics, which of course we would expect since the accuracy of the linearization is a function of the error level itself. The author therefore concludes that the linearized filter is an excellent approximation to the optimal nonlinear filter.

Table 9.2. Three-Dimensional Sensitivity Results

Track Sensor	Maneuver Level	Target	θ (Linearized/ Optimal)	θ (Uncoupled/ Coupled)
A	A (Low)	Closing	0.994	0.956
A	A (Low)	Crossing	0.994	0.961
A	B (High)	Closing	0.998	0.978
A	B (High)	Crossing	0.9985	0.993
E	A (Low)	Closing	0.99984	0.978
E	A (Low)	Crossing	0.999996	0.995
E	B (High)	Closing	0.9999998	0.997
E	B (High)	Crossing	0.9999996	0.9997

Now let us consider the uncoupled linearized filter. This filter is most important since it results in a significant computational advantage. Since the linearized coupled filter is so close to optimal for our problem, we compared the uncoupled filter to it to measure optimality. (Presumably, one might estimate the criterion for this filter relative to the optimal nonlinear filter (if desired) by calculating the product of the two values of θ although the author does not prove this.) We find that the drop in performance is more noticeable for this case but remains within a few percent of the optimal for all cases. Again, for the improved E track sensor, the difference is essentially negligible. The author therefore concludes that, due to the significant computational advantage enjoyed by the uncoupled linearized filter and to the relatively small loss of optimality of this design, that it is to be recommended for implementation. It is also felt that other conditions, such as imperfect target motion modeling and inaccurate knowledge of sensor error statistics, could result in more significant performance reduction from the optimal performance level. This same conclusion has been reached by other researchers such as Cantrell (1973) and, in fact, independent Cartesian filtering is presently employed in all U. S. Navy FCS for major-caliber guns.

E. POLAR AND HYBRID FORMS

As mentioned previously, since the uncoupled Cartesian filter was the one ultimately chosen for implementation, the details of work on the other forms will not be presented here. A brief discussion of the polar and hybrid forms is indicated, however, since they are both interesting to compare and contrast with the Cartesian filters.

The spherical polar state vector is

$$\underline{x}_{p3D} = [r \dot{r} \ddot{r} B \dot{B} \ddot{B} E \dot{E} \ddot{E}]^T \quad (9.22)$$

The kinetic relationship between \underline{x}_{p3D} and \underline{x}_{c3D} can be obtained by doubly differentiating the nonlinear transformation Equations (9.1) and (9.2). The dynamic relationship is then determined by substitution into the three-dimensional Cartesian target motion Equations (9.5-7). As might be expected, this manipulation is quite tedious. The resulting polar dynamic equations are highly nonlinear with rather strong coupling between coordinates. The state extrapolation can be effected exactly however.

Unfortunately, it was not possible for the author or his associates to determine the exact nonlinear extrapolation for the error covariance, and it was necessary to linearize and iterate again in order to construct the optimal nonlinear polar filter. The update process in polar coordinates is, of course, linear and posed no problems. Of course, as far as performance is concerned, the optimal nonlinear filters in both coordinate systems are exactly equivalent. If one were determined to implement an optimal nonlinear filter, then the

choice of coordinate system would be based upon other considerations such as computational burden. This conclusion is essentially identical to that of Monzingo (1972) who decided that both approaches could be made to yield comparable track accuracies.

The basic computational problem centers upon the relative degree of cross coupling and nonlinearity (however one might define such an effect) between the nonlinear update step for the Cartesian filter and the nonlinear extrapolation step for the polar filter. This relative effect is clearly a function of the particular application and would determine the number of iterations required to effect convergence. For example, for long-range tracking of reentry vehicles, Mehra (1971) found that the update nonlinearity was more severe than the dynamic nonlinearity *for the assumed conditions* of that estimation problem. This could be expected since, as mentioned previously, the observation error contours are very distorted at long ranges and the reentry vehicle dynamics does not usually involve high angular rates. The author suspects (but did not demonstrate) that the opposite is the case for our application. (It should be noted also that the observations with the phased-array radar in Mehra's paper were not of angles but of direction cosines for which the nonlinearity is different.) It is also accepted by many, including the author and Monzingo (1972), that the independent (uncoupled) suboptimal polar filter could degrade seriously at close ranges due to the neglect of the "apparent accelerations" mentioned previously.

The concept of the hybrid filter is simply an attempt to minimize the effect of the nonlinearities and to use the best (most linear) parts from both the Cartesian and polar filters. The state vector for this filter is mixed (hence the term "hybrid")

$$\underline{x}_{H3D} = [r \ B \ E \ \dot{x} \ \ddot{x} \ \dot{y} \ \ddot{y} \ \dot{z} \ \ddot{z}]^T \quad (9.23)$$

The only nonlinearity that appears for this filter is with the extrapolation of polar positions since the other state variable extrapolations and the update are linear. The error covariance matrix is still full (9 X 9) and must be linearized for the extrapolation step. Additional work with this filter is required before conclusions can be made but it remains an interesting concept. Variations on this filter have been conceptually explored and include a rotating Cartesian filter with origin maintained at the current estimated target position and orientation such that one axis is aligned with the ray and the orthogonal axis aligned tangent to the sphere in the directions of bearing and elevation. Clearly, additional investigations are needed if one is to make definite conclusions as to the advantages (if any) of these ideas.

X. SUMMARY AND RECOMMENDATIONS

This section is intended to give the reader a brief overview of the studies, results, conclusions, and areas for future work contained in this report. Section I presents the reader a

brief introduction to the fire control problem and the basic considerations attendant to "front-end" (filter/predictor) design. Section II introduces the concepts, notation and equations of the Kalman filter with particular emphasis on reasons for its choice in this application and information required to construct a filter of this type.

The next six Sections (III-VIII) deal exclusively with modeling and parametric behavior for one-dimensional filtering and prediction. The derivative polynomial target model was first explored (Section III) since it is undoubtedly the most common and familiar and parametric behavior of a rather fundamental nature was unavailable for this simple model. Convergence studies indicated that constant jerque and higher-order filters could not be considered for this application—at least as the principal filter—since they take an unacceptable amount of time to adequately settle. Quantitative relationships of filter convergence with polynomial order and observation data rate were determined. The problem of dealing with a maneuvering target with unknown strategy led us to consider the random acceleration target model in Section IV. The author found this model particularly appealing since it statistically recognizes and acknowledges this lack of information and enables one to directly relate the target maneuver parameters (acceleration level and frequency) to the Kalman filter bandwidth and performance matrices. It was demonstrated that this filter converges faster than the constant acceleration filter and, in fact, tends to display behavior somewhere between the constant-acceleration and constant-velocity filters. This model is particularly valuable in that it enables the user to determine fundamental limitations concerning the ability to filter and predict a maneuvering target path. Additional work on target motion analysis, especially from a three-dimensional point of view, is recommended.

Concepts of divergence prevention and adaptive Kalman filtering are considered in Section V. A discussion of divergence and the bandwidth tradeoff is followed by the development and analysis of the use of residual analysis as a maneuver detection tool. Having a divergence detection technique in hand, two concepts of adaptation are developed. The advantage of the dual-bandwidth over the single-variable-bandwidth adaptive filter are presented along with the particular adaptive scheme recommended for implementation. The dual-bandwidth concept is particularly attractive since both narrow and wide bandwidth filters are always in operation and the adaptation algorithm merely selects the output state vector according to the monitored performance of each. Additional work relating to advanced adaptation techniques such as multi-bandwidth parallel or cascade filter banks with hypothesis testing or residual analysis and needed sensitivity adaptive designs are strongly recommended.

The problem of serially correlated observation errors is treated in Section VI. A correlation model and modified Kalman filter and sensitivity algorithm are developed, and results pertaining to observation error statistics are presented.

It would probably not have been possible to actually implement these Kalman filter designs on a real-time, fixed-point mini-computer if it were not for the prefilter and square-root covariance techniques developed in Section VII. The data compression work in particular is critical, and more advanced state-of-the-art techniques should be explored in the future since the payoff in reduced computation is so great with a successful design.

The predictor, of course, is the ultimate product of the system front-end (target sensor plus the filter) so a special section (VIII) was dedicated to the problem of predictor design. After demonstrating that the minimum variance prediction is given by the transition matrix, the sensitivity of the predictor to the choice of acceleration as a constant and prediction accuracy for the limiting case of perfect filtering was discussed.

Finally, in Section IX, the results for the one-dimensional filter work were tied together into the three-dimensional filter design. After developing the optimal nonlinear filter as a standard, it was shown that the linearized observation error covariance approximation is quite accurate for our purposes, and that, by paying only a small cost in performance degradation by neglecting the cross-correlation terms, the three-dimensional filter can be decoupled with a dramatic reduction in required computational load. It is recommended that additional work on hybrid and polar filters be performed along with the target motion analysis problem with the objective of further performance improvement foremost in mind.

The final version of the FORTRAN model incorporating all the features discussed in this report is found in the Appendix.

While it is felt that many improvements, even of a fundamental nature, can be made to this filter design, this model represents the best that the author, with the aid and advice of many associates, could achieve to date. It is believed—or at least fervently hoped—that this model represents an advance over existing filter/predictors currently employed for weapons control purposes.

```

SUBROUTINE DRIVER(IL,BXV,BYV,BZV)
COMMON /FILTER/ DT, MEM, ALPHA, ALPHAS, BET4, BET4S, GAMMA, GAMMA8,
2 QA11,QA11,QA12,QA12,QA13,QA13,QA22,QA22,QA23,QA23,QA33,QA33,QA1,
3 G2,G2,STORE1,STORE2,STORE3,MEN2,MEN3,MEN3,SIGMA,SIGMA,SIGMA,
4 H023,RH023,FH0,FH0,F1,F2,DATA,C.POINTS,SIGMA,SIGMA,SIGMA,
5 TAU1,TAU1,RH012,SIGMA2,SIGMA2,SIGMA2,SIGMA2
REAL MEM2,MEN3,MEN3
COMMON /STATE/ XA1,XA2,XA3,YA1,YA2,YA3,ZA1,ZA2,ZA3,DUMMY(3),
2 XB1,XB2,XB3,YB1,YB2,YB3,ZB1,ZB2,ZB3,DUMMY3(3),
3 SAX11,SAX12,SAX13,SAX22,SAX23,SAX33,SAY11,SAY12,SAY13,SAY22,
4 SAY23,SAY33,SAZ11,SAZ12,SAZ13,SAZ22,SAZ23,SAZ33,SBX11,SBX12,
5 SBX13,SBX22,SBX23,SBX33,SBY11,SBY12,SBY13,SBY22,SBY23,SBY33,
6 SBZ11,SBZ12,SBZ13,SBZ22,SBZ23,SBZ33
REAL MUAXBAR,MUYBAR,MUZBAR,MUAXBAR,MUYBAR,MUZBAR
C AVERAGE MEASUREMENTS
IF (IL.EQ. 1) GO TO 12
ICOUNT = ICOUNT+1
ZX = ZX + BXV/FH0
ZY = ZY + BYV/FH0
ZZ = ZZ + BZV/FH0
IF(ICOUNT.LT. 4) RETURN
13 CONTINUE
C SPECIFY SENSOR STATISTICS
CALL CARTPOL(ZX,ZY,ZZ,R,B,E)
R2 = R**2
SINB2 = (SIN(B))**2
COSB2 = 1.-SINB2
SINE2 = (SIN(E))**2
COSE2 = 1.-SINE2
C PR_FILTER REDUCTIONS
SIGR2 = SIGR2/FH0
SIGB2 = SIGB2/FH0
SIGE2 = SIGE2/FH0
C CARTESIAN STATISTICS
SIGX2 = SINB2*COSE2*SIGR2+R2*COSB2*SIGB2
+R2*SINB2*SINE2*SIGE2
SIGY2 = COSB2*COSE2*SIGR2+R2*SINB2*SIGB2
+R2*COSB2*SINE2*SIGE2
SIGZ2 = SINE2*SIGR2+R2*COSE2*SIGE2
C CALL INDEPENDENT CARTESIAN FILTERS
IF (IL.EQ. 1) GO TO 1
CALL FILTER(XA1,XA2,XA3,SAX11,SAX12,SAX13,SAX22,SAX23,SAX33,
3 ZX,SIGX2,MUAXBAR,MUAXBAR)
2 XB1,XB2,XB3,SBX11,SBX12,SBX13,SBX22,SBX23,SBX33,
2 YB1,YB2,YB3,SBY11,SBY12,SBY13,SBY22,SBY23,SBY33,
CALL FILTER(YA1,YA2,YA3,SAY11,SAY12,SAY13,SAY22,SAY23,SAY33,
2 ZY,SIGY2,MUAYBAR,MUAYBAR)
2 ZB1,ZB2,ZB3,SBZ11,SBZ12,SBZ13,SBZ22,SBZ23,SBZ33,
CALL FILTER(ZA1,ZA2,ZA3,SAZ11,SAZ12,SAZ13,SAZ22,SAZ23,SAZ33,
3 ZZ,SIGZ2,MUAZBAR,MUAZBAR)

```

```

ICOUNT = 0
ZX = 0
ZY = 0
ZZ = 0
RETURN
C INITIALIZE FILTER
1 CONTINUE
C STATE
XA1 = 0.0  SX01 = XA1  SX02 = 0.  SX03 = 0.  SX04 = 0.  SX05 = 0.
YA1 = 0.0  SY01 = YA1  SY02 = 0.  SY03 = 0.  SY04 = 0.  SY05 = 0.
ZA1 = 0.0  SZ01 = ZA1  SZ02 = 0.  SZ03 = 0.  SZ04 = 0.  SZ05 = 0.
C ERROR COVARIANCE SQUARE ROOT
SIGX = SQRT(SIGX2)  SIGY = SQRT(SIGY2)  SIGZ = SQRT(SIGZ2)
SAX11 = STORE3*SIGX  $SAX11 = SAX11
SAY11 = STORE3*SIGY  $SAY11 = SAY11
SAZ11 = STORE3*SIGZ  $SAZ11 = SAZ11
SAX12 = STORE2*SIGX  $SAX12 = SAX12
SAY12 = STORE2*SIGY  $SAY12 = SAY12
SAZ12 = STORE2*SIGZ  $SAZ12 = SAZ12
SAX13 = RH013*SIGX  $SAX13 = SAX13
SAY13 = RH013*SIGY  $SAY13 = SAY13
SAZ13 = RH013*SIGZ  $SAZ13 = SAZ13
SAX22 = STORE1*SIGX  $SAX22 = SAX22
SAY22 = SAX22  $SAY22 = SAY22
SAZ22 = SAX22  $SAZ22 = SAZ22
SAX23 = RH023*SIGX  $SAX23 = SAX23
SAY23 = SAX23  $SAY23 = SAY23
SAZ23 = SAX23  $SAZ23 = SAZ23
SAX33 = SIGX  $SAX33 = SIGX
SAY33 = SIGY  $SAY33 = SIGY
SAZ33 = SIGZ  $SAZ33 = SIGZ
NUAXBAR = 0.  $NUAXBAR = 0.
NUAYBAR = 0.  $NUAYBAR = 0.
NUAZBAR = 0.  $NUAZBAR = 0.
ICOUNT = 0
ZX = 0.
ZY = 0.
ZZ = 0.
RETURN
12 CONTINUE
C CALCULATED CONSTANTS - REQUIRED BY FILTER
DT = FLOAT(MEM)/DATRATE
GAMMAA = EXP(-DT/TAUMA)
BETA = TAUMA*(1.-GAMMAA)
GAMMAB = EXP(-DT/TAUMB)
BETAB = TAUMB*(1.-GAMMAB)
ALPHA = TAUMA**2*(GAMMAA+DT/TAUMA-1.)
ALPHAB = TAUMB**2*(GAMMAB+DT/TAUMB-1.)
QA33 = 2.*DT*SIGMAA**2/TAUMA
QB33 = 2.*DT*SIGMAB**2/TAUMB

```

```

QA23 = DT*QA33/2.
QB23 = DT*QB33/2.
QA22 = 2.*DT*QA23/3.
QB22 = 2.*DT*QB23/3.
QA13 = QA22/2.
QB13 = QB22/2.
QA12 = 3.*DT*QA13/4.
QB12 = 3.*DT*QB13/4.
QA11 = 2.*DT*QA12/5.
QB11 = 2.*DT*QB12/5.
G2 = 1./PCINTS
G1 = 1.-G2
G2 = G2*C**2
STORE1 = SQRT (1.-RHO23**2)
STORE2 = (RHO12-RHO13*RHO23)/STORE1
STORE3 = SQRT (1.-STORE2**2-RHO13**2)
FMEW = MEW
FMEW2 = FMEW**2
SUM = 0.
SUMA = 0.
SUMB = 0.
DO 230 I = 1,MEW
FI = I
SUM = SUM+FI
SUMA = SUM+GAMMAA***(FI/FMEW)
SUMB = SUM+GAMMAB***(FI/FMEW)
230 CONTINUE
SUM = SUM/FMEW2
SUMA = SUMA/(FMEW*GAMMAA)
SUMB = SUMB/(FMEW*GAMMAB)
MEW2 = DT*(1.-SUM)
MEW3 = TAUHA**2*(SUMA-1.)-TAUHB*MEW2
MEW3 = TAUHB**2*(SUMB-1.)-TAUHB*MEW2
F1 = (FMEW+1.)/(2.*FMEW)
F2 = (FMEW-1.)/(2.*FMEW)
SIG3NA = SIGNAM.
SIG3NB = SIGNAMB
ZA = 0XV
ZY = 0YV
ZZ = 0ZV
GO TO 13
END

```

```

SUBROUTINE FILT. R(XA1,XA2,XA3,SA11,SA12,SA13,SA22,SA23,SA33,
2 XB1,XB2,XB3,SB11,SB12,SB13,SB22,SB23,SB33,
3 Z,SIGMA2,NUAB-R,NUAB-R)
C LAST UPDATE - 24 JULY 1974
C ONE-DIMENSIONAL, ADAPTIVE DUAL-BANDWIDTH, KALMAN TRACKING FILTER
C FILTER 1 = NARROW BANDWIDTH
C FILTER 2 = WIDE BANDWIDTH
C VARIABLES
C XA/XB = STATE VECTORS
C 1 = POSITION
C 2 = VELOCITY
C 3 = ACCELERATION
C SA/SB = ERROR COVARIANCE SQUARE ROOT MATRICES
C Z = MEASUREMENT (AVERAGE OF NEW OBSERVATIONS)
C SIGMA2 = VARIANCE OF WHITE MEASUREMENT ERROR
C CONSTANTS
C DT = TIME INCREMENT BETWEEN FILTER CYCLES
COMMON /FILTER/ DT,MEN,ALPHA,ALPHA,ALPHA,BETA,ETA,GAMMA,GAMMA,
2 QA11,Q511,Q412,Q912,Q413,Q913,QA22,Q322,QA23,Q323,QA33,Q833,G1,
3 G2,G2,STORE1,STORE2,STORE3,MEN2,MEN3,MEN3,SIGMA,SIGMA,SIGMA,
4 M023,M023,FH,M,TUP,F1,F2,OUTRATE,C.POINTS,SIGMA,SIGMA,
5 TAMA,TAMA,RM012,SIGMA2,SIGMA2,SIGMA2,SIGMA2
REAL KA1,KA2,KA3,Kb1,Kb2,Kb3,MU1,MU3
REAL MEN2,MEN3,MEN3,NUAB-R,NUAB-R
C STATE EXTRAPOLATION
XA1 = XA1+DT*XA2+ALPHA*XA3
XB1 = XB1+DT*XB2+ALPHA*XB3
XA2 = XA2+3.14*XA3
XB2 = XB2+3.14*XB3
XA3 = GAMMA*XA3
XB3 = GAMMA*XB3
C SQUARE ROOT COVARIANCE EXTRAPOLATION
SIGMA0 = SQRT(SA11**2+SA12**2+SA13**2+SIGMA2)
SIGMA0 = SQRT(SB11**2+SB12**2+SB13**2+SIGMA2)
SA12 = SA12+DT*SA22
SB12 = SB12+DT*SB22
SA13 = SA13+DT*SA23+ALPHA*SA33
SB13 = SB13+DT*SB23+ALPHA*SB33
SA23 = SA23+3.14*SA33
SB23 = SB23+3.14*SB33
SA33 = GAMMA*SA33
SB33 = GAMMA*SB33
M033 = SB33*SQRT(1.+QA33/SB33**2)
M023 = (SB33/M033)*SB23+QA23/M033
M022 = SB22*SQRT(1.+(SB23+M023)*(SB23-M023)+QA22/SB22**2)
M013 = (SE33/M033)*SB13+QA13/M033
M012 = (SB22/M022)*SB12+(SB13*SB23-M013*M023+QA12)/M022
M011 = SB11*SQRT(1.+(SB12+M012)*(SB12-M012)+(SB13+M013)*
2 (SB13-M013)+QA11/SB11**2)
C CORRECT MEASUREMENT FOR TARGET MOTION (WHEN PREFILTERING)

```

```

ZA = Z+MEW2*XA2-MENB3*XA3
ZB = Z+MEW2*XB2-MENB3*XB3
C CALCULATE A PRIORI RESIDUAL
NUA = ZA-XA1
NUB = ZB-XB1
C MANEUVER DETECTOR
PA11 = SA11**2+SA12**2+SA13**2+QA11
PB11 = NB11**2+NB12**2+NB13**2
GA = PA11/SIGMAW2
GB = PB11/SIGMAW2
SIGNUA = SQRT(GA)
SIGNUB = SQRT(GB)
NUABAR = G1*NUA/GA+G2*NUB/SIGNUB
NUBBAR = G1*NUB/GB+G2*NUA/SIGNUB
SWITCH = G2-NUABAR**2
IF (SWITCH .GE. 0.) GO TO 1
C MANEUVER DECLARATION
XA1 = XB1
XA2 = XB2
XA3 = XB3
NUA = NUB
NUABAR = (SIGNUB/SIGNUA)*NUBBAR
NA33 = SA33*SQRT(1.+QB33/SA33**2)
NA23 = (SA33/NA33)*SA23+QB23/NA33
NA22 = SA22*SQRT(1.+((SA23+NA23)*(SA23-NA23)+QB22)/SA22**2)
NA13 = (SA33/NA33)*SA13+QB13/NA33
NA12 = (SA22/NA22)*SA12+(SA13*SA23-NA13*NA23+QB12)/NA22
NA11 = SA11*SQRT(1.+((SA12+NA12)*(SA12-NA12)+(SA13+NA13)*
2 (SA13-NA13)+QB11)/SA11**2)
PA11 = PA11+QB11-QA11
GA = GA-QA11/QB11
GO TO 4
C NO MANEUVER
1 CONTINUE
NA33 = SA33*SQRT(1.+QA33/SA33**2)
NA23 = (SA33/NA33)*SA23+QA23/NA33
NA22 = SA22*SQRT(1.+((SA23+NA23)*(SA23-NA23)+QA22)/SA22**2)
NA13 = (SA33/NA33)*SA13+QA13/NA33
NA12 = (SA22/NA22)*SA12+(SA13*SA23-NA13*NA23+QA12)/NA22
NA11 = SA11*SQRT(1.+((SA12+NA12)*(SA12-NA12)+(SA13+NA13)*
2 (SA13-NA13)+QA11)/SA11**2)
4 CONTINUE
C GAINS
PA12 = NA12*NA22+NA13*NA23
PB12 = NB12*NB22+NB13*NB23
PA13 = NA13*NA33
PB13 = NB13*NB33
KA1 = PA11/GA
KB1 = PB11/GB
KA2 = PA12/GA

```

```

K92 = P012/G0
KA3 = PA13/GA
KB3 = P013/G0
C UPDATE STATE
FACTORA = 1./ (F1+F2*SIGNUA0/SIGNUA)
FACTORB = 1./ (F1+F2*SIGNUB0/SIGNUB)
NUA = FACTORA*NUA
NUB = FACTORB*NUB
XA1 = XA1+KA1*NUA
XB1 = XB1+KB1*NUB
XA2 = XA2+KA2*NUA
XB2 = XB2+KB2*NUB
XA3 = XA3+KA3*NUA
XB3 = XB3+KB3*NUB
C UPDATE COVARIANCE SQUARE ROOT
SA33 = WA33*SQRT (1.-GA*KA3*KA3/WA33**2)
SB33 = WB33*SQRT (1.-GB*KB3*KB3/WB33**2)
SA23 = (WA33/SA33)*WA23-GA*KA2*KA3/SA33
SB23 = (WB33/SB33)*WB23-GB*KB2*KB3/SB33
SA22 = WA22*SQRT (1.+((WA23+SA23)*(WA23-SA23)-GA*KA2*KA2)/WA22**2)
SB22 = WB22*SQRT (1.+((WB23+SB23)*(WB23-SB23)-GB*KB2*KB2)/WB22**2)
SA13 = (WA33/SA33)*WA13-GA*KA1*KA3/SA33
SB13 = (WB33/SB33)*WB13-GB*KB1*KB3/SB33
SA12 = (WA22/SA22)*WA12+(WA13*WA23-SA13*SA23-GA*KA1*KA2)/SA22
SB12 = (WB22/SB22)*WB12+(WB13*WB23-SB13*SB23-GB*KB1*KB2)/SB22
SA11 = WA11*SQRT (1.+((WA12+SA12)*(WA12-SA12)+(WA13+SA13)*
2 (WA13-SA13)-GA*KA1*KA1)/WA11**2)
SB11 = WB11*SQRT (1.+((WB12+SB12)*(WB12-SB12)+(WB13+SB13)*
2 (WB13-SB13)-GB*KB1*KB1)/WB11**2)
RETURN
END

```

```

SUBROUTINE PREDICI(X,TP)
COMMON /FILTER/ OT,MEN,ALPHA,ALPHA0,BETA0,BETA2,GAMMA,GAMMA0,
2 GA11,Q311,WA12,Q312,QA13,GA13,GA22,Q322,GA23,Q323,QA33,Q333,G1,
3 G2,G2,STOR1,STORE2,STORE3,MEN2,MEN3,ENH3,SIGNA,SIGNB,SIGNC,
4 MO23,KMO23,FH,W,TAUP,F1,F2,DATAT,C.POINTS,SIGMAH,SIGMANO,
5 T-UP,T-UP0,RHO12,SIGMA2,SIGMA2,SIGMA2,SIGMA2,
DIMENSION X(12)
ALPHA = T-UP**2*(EXP(-TP/T-UP)+TP/T-UP-1.)
X(10) = X(1)+TP*X(2)+ALPHA*X(3)
X(11) = X(4)+TF*X(5)+ALPHA*X(6)
X(12) = X(7)+TF*X(8)+ALPHA*X(9)
RETURN
END

```


REFERENCES

- Alspach, D. L. (1973), "A Parallel Processing Solution to the Adaptive Kalman Filtering Problem with Vector Measurements," *Comput. and Elect. Engng*, 1, 83-94.
- Balas, L. F. (1967) "Digital Simulation of Band-Limited White Gaussian Noise," Naval Electronics Laboratory Center Technical Memo.
- Bendat, J. S. and A. G. Piersol (1971), *Random Data: Analysis and Measurement Procedures* (Wiley-Interscience, New York).
- Brown, C. M. and C. F. Price (1974), "Adaptive Tracking Filter Design and Evaluation for Gun Fire Control Systems," The Analytic Sciences Corporation TR-381-1 (Naval Ordnance Systems Command Contract No. N0017-73-C-4323).
- Bryson, A. E., Jr. and L. J. Hendrikson (1968), "Estimation Using Sampled Data Containing Sequentially Correlated Noise," *Journal of Spacecraft* 5, 662-665.
- Cantrell, B. H. (1973), "Description of an Alpha-Beta Filter in Cartesian Coordinates," Naval Research Laboratory, NRL Report 7548.
- Carison, N. A. (1973), "Fast Triangular Formulation of the Square Root Filter," *AIAA Journal* 11, 1259-1265.
- Clark, B. L. (1973), "A Predictor Based on a Nondeterministic Model," An unpublished paper of the Warfare Analysis Department, Naval Weapons Laboratory, Dahlgren, Virginia. (To be published as part of an NWL Technical Report on new prediction techniques.)
- D'Appolito, J. A. (1971), "The Evaluation of Kalman Filter Designs for Multisensor Integrated Navigation Systems," Technical Report AFAL-TR-70-271.
- Demetry, J. S. and H. A. Titus (1968), "Adaptive Tracking of Maneuvering Targets," *IEEE Transactions on Automatic Control* AC-13, 749-750.
- Epstein, G. (1971), "Adaptive Memory Trackers," Fall Joint Computer Conference 663-668.
- Friedland, B. (1969), "Treatment of Bias in Recursive Filtering," *IEEE Transactions on Automatic Control* AC-14, 359-367.
- Gelb, A. (ed) (1974), *Applied Optimal Estimation* (MIT Press, Cambridge, Mass.).
- Hagar, H. (1973), "Model Error Compensation Techniques for Linear Filtering," Ph.D. Thesis, University of Texas at Austin.
- Hampton, R. L. and J. R. Cooke (1973), "Unsupervised Tracking of Maneuvering Vehicles," *IEEE Transactions on Aerospace and Electronic Systems* AES-9, 197-207.
- Jazwinski, A. H. (1969), "Adaptive Filtering," *Automatica* 5, 475-485.
- Jazwinski, A. H. (1970), *Stochastic Processes and Filtering Theory* (Academic Press, New York).
- Kalman, R. E. (1960), "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME J. Basic Engineering* 82D, 35-45.
- Kalman, R. E. (1961), "New Results in Linear Filtering and Prediction Theory," *Transactions of the ASME J. Basic Engineering* 83D, 95-108.

- Kaminski, P. G., A. E. Bryson and S. F. Schmidt (1971), "Discrete Square Root Filtering: A Survey of Current Techniques," *IEEE Transactions on Automatic Control* AC-16, 727-736.
- McAulay, R. J. and E. Denlinger (1973), "Decision-Directed Adaptive Tracker," *IEEE Trans on Aerospace and Electronic Systems* AES-9, 229-236.
- Mehra, R. K. (1972), "Approaches to Adaptive Filtering," *IEEE Trans on Automatic Control* AC-17, 693-698.
- Mehra, R. K. (1971, #1), "On-Line Identification of Linear Dynamics Systems with Application to Kalman Filtering," *IEEE Transactions on Automatic Control* AC-16, 12-21.
- Mehra, R. (1970), "On the Identification of Variances and Adaptive Kalman Filtering," *IEEE Transactions on Automatic Control* AC-15, 175-184.
- Mehra, R. K. (1971, #2), "A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking," *IEEE Transactions on Automatic Control* AC-16, No. 4, 307-319.
- Monzingo, R. A. (1972), "IPD/TAS Tracking Coordinate Study," Improved Point Defense/Target Acquisition System Engineering Notebook, Book 1, Section 15, Seq. 1-1, Hughes Aircraft Company.
- Moose, R. L. (1972), "An Adaptive Estimator for Passive Range and Depth Determination of a Maneuvering Target," Naval Underwater Systems Center Technical Report 4375.
- Morrison, N. (1969), *Introduction to Sequential Smoothing and Prediction* (McGraw-Hill, New York).
- Nahi, N. E. and B. M. Schaefer (1972), "Decision-Directed Adaptive Recursive Estimators: Divergence Prevention," *IEEE Transactions on Automatic Control* AC-17, 61-68.
- Potter, J. E. (1963), "New Statistical Formulas," Space Guidance Analysis Memo 40, Instrumentation Lab., MIT, Cambridge, Mass.
- Sage, A. P. and J. L. Melsa (1971), *Estimation Theory with Applications to Communications and Control* (McGraw-Hill, New York).
- Schmidt, S. F. (1970), "Computational Techniques in Kalman Filtering," Chapter 3 of *Theory and Applications of Kalman Filtering*, NATO Advisory Group for Aerospace and Development, AGARDograph 139.
- Sidar, M. and Y. Bar-Shlomo (1972), "A Comparison of the Effectiveness of Some Adaptive Optimal Filtering Techniques Applied to the Gyrocompassing Problem." Paper submitted to the 5th IFAC Congress, Paris.
- Singer, R. A. and K. W. Behnke (1971), "Real-Time Tracking Filter Evaluation and Selection for Tactical Applications," *IEEE Transactions on Aerospace and Electronic Systems* AES-7, 100-110.
- Singer, R. A. (1970), "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems* AES-6, 473-483.
- Soeda, T. and T. Yoshimura (1973), "A Practical Filter for Systems with Unknown Parameters," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 396-401.
- Sorenson, H. W. (1966), "Kalman Filtering Techniques," *Advanced Control Systems* 3.

- Thorp, J. S. (1973), "Optimal Tracking of Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems* AES-9, 512-519.
- Weiss, I. M. (1970), "A Survey of Discrete Kalman-Bucy Filtering with Unknown Noise Covariances," AIAA Paper No. 70-955.
- Wu, T. K. (1973), "Application of Kalman Filtering to the Tactical Aircraft Navigation System," Presented at the Symposium on the Application of Control Theory to Modern Weapon Systems, California City, Cal. Sponsored by the Naval Weapons Center.

APPENDIX A

Listings of Computer Programs

KALMAN -- This subroutine performs the matrix algebra involved in one cycle of the general Kalman filter. The general Kalman filter assumes a completely linear system with uncorrelated measurement and process noise. The equations are summarized in Table 2.1 and reproduced here in the order in which they are used in the subroutine. Notice that we have included a deterministic forcing function in the state equation. This inclusion has no effect on the remaining equations.

$$\hat{X}(k/k-1) = \phi(k, k-1) \hat{X}(k-1/k-1) + f(k-1) \quad (A.1)$$

$$P(k/k-1) = \phi(k, k-1) P(k-1/k-1) \phi^T(k, k-1) + Q(k-1) \quad (A.2)$$

$$K(k) = P(k/k-1) H^T(k) [H(k) P(k/k-1) H^T(k) + R(k)]^{-1} \quad (A.3)$$

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K(k) [Z(k) - H(k) \hat{X}(k/k-1)] \quad (A.4)$$

$$P(k/k) = [I - K(k) H(k)] P(k/k-1) \quad (A.5)$$

```

SUBROUTINE KALMAN(X,PHI,F,P,Q,R,C,D,N,M,NDIM,NTIME)
C LAST UPDATE ON SEPTEMBER 1973
C GENERAL KALMAN FILTER CYCLE FOR LINEAR SYSTEM WITH WHITE MEASUREMENT
C AND PROCESS NOISE. GIVEN INPUT AT TIME(K-1), THIS ROUTINE RETURNS
C OUTPUT AT TIME(K).
C X - STATE VECTOR - INPUT(K-1) -> OUTPUT(K)
C PHI - TRANSITION MATRIX - INPUT
C F - DETERMINISTIC FORCING VECTOR - INPUT
C P - ERROR COVARIANCE MATRIX - INPUT(K-1) -> OUTPUT(K)
C Q - PROCESS NOISE MATRIX - INPUT
C R - MEASUREMENT NOISE MATRIX - INPUT
C H - OBSERVATION MATRIX - INPUT
C K - GAIN MATRIX - CALCULATED
C Z - MEASUREMENT VECTOR - INPUT
C A,B,C,D - WORKING STORAGE
C NDIM,NTIME - DIMENSIONS OF VARIABLES IN CALLING ROUTINE
C N,M - OPERATING DIMENSIONS
C NOTE - C MUST BE DIMENSIONED M BY M FOR THE MATRIX INVERSION TO WORK
      REAL X
      DIMENSION X(NDIM),PHI(NDIM,NDIM),F(NDIM),P(NDIM,NDIM),Q(NDIM,NDIM)
      Z(NDIM),R(NDIM,NDIM),K(NDIM,NDIM),H(NDIM,NDIM),C(NDIM),
      D(NDIM,NDIM),D(NDIM,NDIM),D(NDIM,NDIM)
C EXTRAPOLATION
C A = PHI(K,K-1)*X(K-1/K-1)+F(K-1)
C B = PHI(K,K-1)*P(K-1/K-1)*TRANSPOSE(PHI)+Q(K-1)
      DO 1 I = 1,N
        A(I) = F(I)
        DO 1 J = 1,N
          A(I) = A(I)+PHI(I,J)*X(J)
          B(I,J) = PHI(I,J)
        DO 1 KY = 1,N
          DO 1 L = 1,N
            B(I,J) = B(I,J)+PHI(I,KY)*PHI(J,L)*P(KY,L)
          CONTINUE

```

```

C K(K/K-1) = A
C P(K/K-1) = B
  DO 2 I = 1,M
    Y(I) = A(I)
  DO 2 J = 1,M
    P(I,J) = P(I,J)
  CONTINUE
C UPDATE
C C = H(K)*P(K/K-1)*TRANSPOSE(H(K))+P(K)
  DO 3 I = 1,M
    DO 3 J = 1,M
      C(I,J) = P(I,J)
  DO 3 KK = 1,M
    DO 3 L = 1,M
      C(I,J) = C(I,J)+H(I,KK)*H(J,L)*P(KK,L)
  CONTINUE
C D = P(K/K-1)*TRANSPOSE(H(K))
  DO 4 I = 1,M
    DO 4 J = 1,M
      D(I,J) = 0
  DO 4 KK = 1,M
    D(I,J) = D(I,J)+P(I,KK)*H(J,KK)
  CONTINUE
C C = INVERSE(C)
  IF (M.GV. 1) CALL MATOP(X(10,M,M,0,0,C,M,DET)
  IF (M.GV. 1) C(1,1) = 1./C(1,1)
C K(K) = 0
  DO 5 I = 1,M
    DO 5 J = 1,M
      K(I,J) = 0
  DO 5 KK = 1,M
    K(I,J) = K(I,J)+D(I,KK)*C(KK,J)
  CONTINUE
C C(I,1) = Y(K)-H(K)*Y(V/K-1)
  DO 6 I = 1,M
    C(I,1) = Y(I)
  DO 6 J = 1,M
    C(I,1) = C(I,1)-H(I,J)*Y(J)
  CONTINUE
C Y(K/K) = Y(K/K-1)+K(K)*C(I,1)
  DO 7 I = 1,M
    DO 7 J = 1,M
      Y(I) = Y(I)+K(I,J)*C(J,1)
  CONTINUE
C Q = K(K)*H(K)*P(K/K-1)
  DO 8 I = 1,M
    DO 8 J = 1,M
      Q(I,J) = 0
  DO 8 KK = 1,M
    DO 8 L = 1,M
      Q(I,J) = Q(I,J)+K(I,KK)*H(KK,L)*P(L,J)
  CONTINUE
C P(V/K) = P(K/K-1)-Q
  DO 9 I = 1,M
    DO 9 J = 1,M
      P(I,J) = P(I,J)-Q(I,J)
  CONTINUE
  PRINT
  END

```



```

      SUBROUTINE CORNUN(XNOISE,SIGMA,TAU,DELTAT,I,N,RHO,SIGMAW)
C THIS SUBROUTINE GENERATES THE I-TH VALUE OF A SEQUENCE OF AUTO-
C CORRELATED RANDOM NUMBERS WITH ZERO MEAN
C EQUIVALENT TO A LOW-PASS FILTER WITH CUTOFF FREQUENCY =  $1/(2\pi\tau)$ 
C HERTZ DRIVEN BY WHITE GAUSSIAN NOISE
C XNOISE - OUTPUT NEW RANDOM NUMBER - INPUT OLD VALUE
C SIGMA - STANDARD DEVIATION OF OUTPUT SEQUENCE
C TAU - EXPONENTIAL AUTOCORRELATION TIME CONSTANT
C DELTAT - CONSTANT SAMPLING INTERVAL
C I - INDEX OF SEQUENCE (.GE. 1)
C THE FOLLOWING QUANTITIES ARE COMPUTED INITIALLY (I = 1) AND MUST BE
C STORED IN THE CALLING ROUTINE FOR USE WHEN I .GT. 1
C N - FACTOR OF INCREASE OF EFFECTIVE SAMPLING
C RHO - CORRELATION COEFFICIENT AT EFFECTIVE SAMPLING RATE
C SIGMAW - STANDARD DEVIATION OF WHITE-NOISE INPUT
C INITIALIZE GENERATOR
      IF (I .GT. 1) GO TO 1
      CALL RANDOM(RANNUM,1)
      XNOISE = SIGMA*RANNUM
C FIND SAMPLING TIME TO SATISFY SHANNON SAMPLING THEOREM
      IF ((DELTAT/TAU) .GT. 10.) GO TO 3
      N = 1.414.*DELTAT/TAU
      RHO = EXP(-DELTAT/(TAU*FLUAT(N)))
      SIGMAW = SIGMA*SQRT(1.-RHO**2)
      RETURN
3 CONTINUE
      N = 1
      RHO = 0.
      SIGMAW = SIGMA
      RETURN
1 CONTINUE
      DO 2 J = 1,N
      CALL RANDOM(RANNUM,1)
      XNOISE = RHO*XNOISE + SIGMAW*RANNUM
2 CONTINUE
      RETURN
      END

```


CORKAL - This subroutine is the general (matrix) version of the Kalman filter which has been restructured to accept serially correlated measurement error. The model and algorithm development are presented in Section VI. The matrix equations which this algorithm computes are shown in Table 6.2.

```

SUBROUTINE CORKAL(X,PHI,F,P,Q,H,HOLD,PSI,Z,ETA,K,N,M,IOUT)
C LAST UPDATE - 1 NOVEMBER 1973
C GENERAL KALMAN FILTER CYCLE FOR LINEAR SYSTEM WITH AUTOCORRELATED
C MEASUREMENT NOISE - GIVEN INPUT AT TIME (K-1), THIS ROUTINE RETURNS
C OUTPUT AT TIME (K) - MATRICES ARE NOT REDEFINED EXCEPT WHERE
C INDICATED
C X(N) - STATE VECTOR - INPUT(K-1) - OUTPUT(K)
C PHI(N,M) - TRANSITION MATRIX - INPUT(K,K-1)
C F(M) - DETERMINISTIC FORCING VECTOR - INPUT(K-1)
C P(N,N) - ERROR COVARIANCE MATRIX - INPUT(K-1/K-1) - OUTPUT(K/K)
C Q(N,N) - PROCESS NOISE MATRIX - INPUT(K-1)
C R(M,M) - MEASUREMENT ERROR MATRIX - WHITE NOISE PORTION - INPUT(K-1)
C H(N,M) - OBSERVATION MATRIX - CURRENT - INPUT(K)
C HOLD(M,M) - OBSERVATION MATRIX - LAST TIME - H(K-1) - INPUT(K-1)
C PSI(M,M) - AUTOCORRELATION TRANSITION MATRIX - INPUT(K,K-1)
C Z(M) - MEASUREMENT VECTOR - INPUT(K)
C ETA(M) - PRODUCT OF PSI(K,K-1) TIMES THE A POSTERIORI RESIDUAL -
C INPUT(K-1) - OUTPUT(K)
C K(N,M) - GAIN MATRIX - CALCULATED - OUTPUT(K)
C N - SIZE OF X
C M - SIZE OF Z
C IOUT - OUTPUT SWITCH (SET TO 1 TO JUMP ALL INTERNAL OUTPUT)
C A(N),B(N,M),C(M,M),D(N,M) - WORKING SPACE
C NOTE - (N,M) MUST NOT EXCEED (10,7) - IF EITHER DOES, CHANGE THE
C DIMENSIONS OF THE WORKING SPACE AND THE NEXT-TO-LAST ARGUMENT
C OF MATRIX INVERSION
REAL *
DIMENSION X(N),PHI(N,M),F(M),P(N,N),Q(N,N),R(M,M),H(N,M),
2 HOLD(M,M),PSI(M,M),Z(M),ETA(M),K(N,M)
DIMENSION A(10),B(10,10),C(3,3),D(10,3)
C EXTRAPOLATION
C 1 = PHI*X*F
C 3 = PHI*P*TRANSPOSE(PHI)*Q
DO 1 I = 1,N
A(I) = F(I)
DO 1 J = 1,M
A(I) = A(I)+PHI(I,J)*F(J)
B(I,J) = C(I,J)
DO 1 KK = 1,N
DO 1 L = 1,M
B(I,J) = B(I,J)+PHI(I,KK)*P(KK,L)*PHI(J,L)
1 CONTINUE
C C = P*H*H*TRANSPOSE(H)+PSI*HOLD*H*TRANSPOSE(HOLD)*TRANSPOSE(PSI)
C -H*PHI*P*TRANSPOSE(HOLD)*TRANSPOSE(PSI)
C -PSI*HOLD*P*TRANSPOSE(PHI)*TRANSPOSE(H)
DO 2 I = 1,M
DO 2 J = 1,M
C(I,J) = B(I,J)
DO 2 KK = 1,N
DO 2 L = 1,M

```

```

      C(I,J) = C(I,J)+H(I,KK)*B(KK,L)*H(J,L)
      DO 2 M = 1,M
      DO 9 NN = 1,N
      C(I,J) = C(I,J)+PSI(I,MM)*HOLD(MN,L)*P(L,KK)*HOLD(MM,KK)*PSI(J,MM)
3 CONTINUE
      DO 10 MM = 1,M
      C(I,J) = C(I,J)-H(I,KK)*PHI(KK,L)*P(L,NN)*HOLD(MM,NN)*PSI(J,MM)
      2 -PSI(I,MM)*HOLD(MM,KK)*P(KK,L)*PHI(MN,L)*HOLD(J,NN)
10 CONTINUE
      2 CONTINUE
      C = 3*TRANSPOSE(H)-PHI*P*TRANSPOSE(HOLD)*PSI
      DO 3 I = 1,M
      DO 3 J = 1,M
      D(I,J) = C.
      DO 3 KK = 1,N
      D(I,J) = D(I,J)+R(I,KK)*H(J,KK)
      DO 3 L = 1,N
      DO 3 MM = 1,M
      D(I,J) = D(I,J)-P*H(I,KK)*P(KK,L)*HOLD(MM,L)*PSI(MM,J)
3 CONTINUE
C GAIN
      C = INVERSE(C)
      IF (M .GT. 1) CALL MATRIN(10,P,M,N,C,J,DET)
      IF (M .EQ. 1) C(1,1) = 1./C(1,1)
      C = 3*C
      DO 4 I = 1,M
      DO 4 J = 1,M
      K(I,J) = C.
      DO 4 KK = 1,N
      K(I,J) = K(I,J)+D(I,KK)*C(KK,J)
4 CONTINUE
C UPDATE
      C = A*(2-H*A-E*A)
      DO 5 I = 1,M
      X(I) = A(I)
      DO 5 J = 1,M
      Y(I) = X(I)-K(I,J)*(Z(J)-ETA(J))
      DO 5 KK = 1,N
      X(I) = X(I)+K(I,J)*H(J,KK)*A(KK)
5 CONTINUE
      C = E-J*K
      DO 6 I = 1,M
      DO 6 J = 1,M
      P(I,J) = 3(I,J)
      DO 6 KK = 1,N
      P(I,J) = P(I,J)-3(I,KK)*K(J,KK)
6 CONTINUE
      C = PSI*(2-H*X)
      DO 7 I = 1,M
      ETA(I) = C.

```

```

DO 7 J = 1,M
  ETA(I) = ETA(I)+PSI(I,J)*Z(J)
DO 7 KK = 1,M
  ETA(I) = ETA(I)-PSI(I,J)*H(J,KK)*V(KK)
7 CONTINUE
IF(IOUT.EQ. 3) RETURN
WRITE(6,11)
11 FORMAT(' EXTRAPOLATED STATE ')
WRITE(6,8) (A(I), I = 1,N)
WRITE(6,12)
12 FORMAT(' EXTRAPOLATED COVARIANCE ')
WRITE(6,9) ((R(I,J), J = 1,M), I = 1,N)
WRITE(6,13)
13 FORMAT(' GAINS ')
WRITE(6,8) (K(I,J), J = 1,M), I = 1,N)
WRITE(6,14)
14 FORMAT(' UPDATED STATE ')
WRITE(6,8) (X(I), I = 1,N)
WRITE(6,15)
15 FORMAT(' UPDATED COVARIANCE ')
WRITE(6,8) ((P(I,J), J = 1,M), I = 1,N)
6 FORMAT(10F3.5)
RETURN
END

```

DISTRIBUTION

Defense Documentation Center (12)
Cameron Station
Alexandria, VA 22314

Naval Sea Systems Command
Washington, DC 20362
ATTN: NOP (Capt. T. L. Meeks) (3)
NSEA-653 (Capt. E. J. Kirschke, Cdr. W. Nelson, F. Rule)
NSEA-031
NSEA-06C2 (R. Hu)
NSEA-6513 (LCDR W. Robbins)

Superintendent
Naval Postgraduate School
Monterey, CA 93940
ATTN: Dr. H. Titus, Mechanical Engineering Dept., Library (3)

Ballistic Research Laboratory
U.S. Army Aberdeen Research and Development Center
Aberdeen Proving Ground
Aberdeen, MD 21005
ATTN: Library

Commander
Air Force Eastern Test Range
Patrick Air Force Base, FL 32925
ATTN: Mr. M. Kennedy, Bldg 981
Range Measurements Lab

Commander
Eglin Air Force Base, FL 32542
ATTN: Mr. Ron While, Bldg 380
Computer Sciences Lab

Commander
Naval Electronic Systems Command
Washington, DC 20360
ATTN: Mr. S. Grogan

DISTRIBUTION--(Continued)

**Commander
Naval Weapons Center
China Lake, CA 93555**

**Naval Weapons Station
Fleet Analysis Center
Seal Beach
Corona, CA 91720
ATTN: T. Ferryman (Code 8423)
S. Hilker (Code 8423)**

**Vitro Laboratories
3471 Saviers Rd.
Oxnard, CA 93030
ATTN: E. E. Foytek**

**Chief of Naval Operations
Department of the Navy
Washington, DC 20350
ATTN: OP 98**

**Commander
Naval Air Systems Command
Washington, DC 20360**

**Director of Defense Research and Engineering
Washington, DC 20390**

**Director
Naval Research Laboratory
Washington, DC 20390
ATTN: Technical Information Division**

**Commanding Officer
Naval Weapons Evaluation Facility
Kirkland Air Force Base
Albuquerque, NM 87117**

DISTRIBUTION--(Continued)

**Commanding Officer
Operational Test and Evaluation Force
Naval Air Station
Norfolk, VA 23511**

**Weapon System Analysis
Quantico Marine Corps
Quantico, VA 22134**

**Superintendent
U.S. Naval Academy
Annapolis, MD 21402
ATTN: Library**

**Applied Physics Laboratory
The John Hopkins University
8621 Georgia Avenue
Silver Spring, MD 20910**

**Air Force Weapons Laboratory
Kirkland Air Force Base, NM 87117**

**Sandia Laboratories
Box 5800
Albuquerque, NM 87115
ATTN: Mr. Marv Bander**

**Director
Defense Intelligence Agency
Washington, DC 20301**

**Institute for Defense Analysis
400 Army-Navy Drive
Arlington, VA 22202**

**Library of Congress
Washington, DC 20540
ATTN: Gift and Exchange Division**

DISTRIBUTION--(Continued)

Local:

CL-30
DF
DF-10
DF-20
DF-30
DF-32 (Fontenot)
DF-34 (Dorsey)
DF-36 (Leong)
DF-36 (Crisp)
DF-36 (Diehl)
DF-36 (Frazer)
DF-36 (McCloud)
DG
DG-04 (Chadwick)
DG-10 (Clodius)
DG-10 (Logan)
DG-10 (McCants)
DG-10 (Greenwald)
DG-10 (Weddell)
DG-40
DG-40 (Jones)
DK
DK-02 (Cohen)
DK-10
DK-13 (Young)
DK-20
DK-40 (Brown)
DK-50 (Alexander)
DK-51 (Hall)
DK-55 (Chun)
DN
DN-10
DN-20 (Miller)
DN-20 (Clark)
DN-20
DN-23 (Robinson)
DN-23 (Price)

DISTRIBUTION--(Continued)

Ferranti Ltd.
Digital Systems Division
Western Road, Bracknell
Berkshire, England
ATTN: Dr. John Moon

American Defense Preparedness Association
819 Union Trust Building
740 15th Street, N.W.
Washington, DC 20005

Office of Naval Research
Dept. of the Navy
Washington, DC 20360
ATTN: Mr. M. Cooper

School of Aero/Mech. Sciences
Upson Hall
Cornell University
Ithaca, NY 14850

KTECH Corporation
911 Pennsylvania Ave.
Albuquerque, NM 87110

ITT Gilfillan
7821 Orron Ave.
P.O. Box 7713
Van Nuys, CA 91409

Raytheon Corporation
Hartwell Road
Bedford, MA 01730
ATTN: Dr. D. V. Stallard
Mail Stub S1-14

DISTRIBUTION--(Continued)

Local:--(Continued)

DN-23 (Auger)
DN-23 (Lindsay)
DN-30 (Straub)
DN-30 (Marier)
DX-21
DX-222
DX-40
WX-40

(2)

(6)